



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Comparison and coupling of continuous and hybridizable discontinuous Galerkin methods : application to multi-physics problems

Mahendra Paipuri

ADVERTIMENT La consulta d'aquesta tesi queda condicionada a l'acceptació de les següents condicions d'ús: La difusió d'aquesta tesi per mitjà del repositori institucional UPCommons (<http://upcommons.upc.edu/tesis>) i el repositori cooperatiu TDX (<http://www.tdx.cat/>) ha estat autoritzada pels titulars dels drets de propietat intel·lectual **únicament per a usos privats** emmarcats en activitats d'investigació i docència. No s'autoritza la seva reproducció amb finalitats de lucre ni la seva difusió i posada a disposició des d'un lloc aliè al servei UPCommons o TDX. No s'autoritza la presentació del seu contingut en una finestra o marc aliè a UPCommons (*framing*). Aquesta reserva de drets afecta tant al resum de presentació de la tesi com als seus continguts. En la utilització o cita de parts de la tesi és obligat indicar el nom de la persona autora.

ADVERTENCIA La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso: La difusión de esta tesis por medio del repositorio institucional UPCommons (<http://upcommons.upc.edu/tesis>) y el repositorio cooperativo TDR (<http://www.tdx.cat/?locale-attribute=es>) ha sido autorizada por los titulares de los derechos de propiedad intelectual **únicamente para usos privados enmarcados** en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio UPCommons. No se autoriza la presentación de su contenido en una ventana o marco ajeno a UPCommons (*framing*). Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

WARNING On having consulted this thesis you're accepting the following use conditions: Spreading this thesis by the institutional repository UPCommons (<http://upcommons.upc.edu/tesis>) and the cooperative repository TDX (<http://www.tdx.cat/?locale-attribute=en>) has been authorized by the titular of the intellectual property rights **only for private uses** placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized neither its spreading nor availability from a site foreign to the UPCommons service. Introducing its content in a window or frame foreign to the UPCommons service is not authorized (*framing*). These rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it's obliged to indicate the name of the author.

COMPARISON AND COUPLING OF CONTINUOUS AND HYBRIDIZABLE DISCONTINUOUS GALERKIN METHODS: APPLICATION TO MULTI-PHYSICS PROBLEMS

Mahendra Paipuri



Doctoral Thesis

Advisor: Carlos Manuel Tiago Tavares Fernandes

Co-advisor: Sonia Fernández-Méndez

Departament d'Enginyeria Civil i Ambiental

Erasmus Mundus PhD program Simulation in Engineering and
Entrepreneurship Development (SEED)

Barcelona, December, 2017

Abstract

This thesis proposes a coupled continuous and hybridizable discontinuous Galerkin formulation to solve conjugate heat transfer problems. This model is then used to find the thermal response of Glass Fiber Reinforced Polymer (GFRP) tubular cross-section under fire.

The first step of this thesis is to compare the computational efficiency of high-order Continuous Galerkin (CG) and Hybridizable Discontinuous Galerkin (HDG) methods for incompressible fluid flow problems in low Reynolds number regimes. Only 2-D examples and direct solvers are considered in the present work. A thoroughly comparison in terms of CPU time and accuracy for both discretization methods is made under the same platform. Various results presented suggests that HDG can be more efficient than CG when the CPU time, for a given degree, is considered. The stability of HDG and CG is studied using a manufactured solution that produces a sharp boundary layer, confirming that HDG provides smooth converged solutions in the presence of sharp fronts whereas, CG failed to converge due to the presence of numerical oscillations.

Following, the solution of the coupled Navier–Stokes/convection-diffusion problem, using Boussinesq approximation, is formulated within the HDG framework and analysed using numerical experiments and benchmark problems. A coupling strategy between HDG and CG methods is proposed in the framework of second-order elliptic operators. The coupled formulation is implemented and its convergence properties are established numerically by using manufactured solutions. Finally, the proposed coupled formulation between HDG and CG for heat equation is combined with the coupled Navier–Stokes/convection diffusion equations to formulate a new CG-HDG model for solving conjugate heat transfer problems. Benchmark examples are solved using the proposed model and validated with literature values.

The final part of the thesis applies the proposed CG-HDG coupled formulation to predict the thermal response of the GFRP tubular cross-section. The radiosity equation that governs the internal radiation is added to the CG-HDG coupled model. Estimates of the discretization errors are computed in order to establish the confidence intervals for quantities of interest. Results with the geometry having curved corners in the cavity are presented and shown to be within the estimated uncertainty intervals. CPU times for the linear solver suggests that the proposed CG-HDG model is more efficient than CG-CG model in all the cases considered.

keywords: hybridizable discontinuous Galerkin, coupling, conjugate heat transfer, GFRP, computational efficiency.

Resumo

Neste trabalho é proposta uma formulação para acoplar os modelos *continuous* e *hybridizable discontinuous Galerkin* a fim de analisar problemas conjugados de transferência de calor. Este modelo é então usado para estudar a resposta térmica de perfis pultrudidos de secção tubular em polímero reforçado com fibras de vidro (GFRP) sob a acção do fogo.

O primeiro passo desta tese é comparar a eficiência computacional dos métodos *Continuous Galerkin* (CG) e *Hybridizable Discontinuous Galerkin* (HDG) de elevada ordem para problemas de escoamento de fluidos incompressíveis para valores reduzidos do número Reynolds. Apenas exemplos bidimensionais e métodos directos são considerados no presente trabalho. Uma comparação exhaustiva em termos de tempo de CPU e precisão para ambos os métodos de discretização é efectuada sob uma plataforma comum. Os resultados apresentados sugerem que, em termos do tempo de CPU requerido, o HDG pode ser mais eficiente que o CG, para um determinado grau. A estabilidade do HDG e CG é estudada usando uma solução fabricada que produz uma abrupta descontinuidade, confirmando que o HDG fornece soluções convergentes e suaves na presença de descontinuidades, enquanto o CG não conseguiu convergir devido à presença de oscilações numéricas.

Em seguida, a solução do problema acoplado Navier-Stokes/convecção-difusão, utilizando a aproximação de Boussinesq, é formulada no contexto HDG e analisada usando soluções de referência. Uma estratégia de acoplamento entre os métodos HDG e CG é proposta no âmbito de operadores elípticos de segunda ordem. A formulação acoplada é implementada e suas propriedades de convergência são estabelecidas numericamente usando soluções fabricadas. Finalmente, a formulação acoplada proposta entre HDG e CG para a equação do calor é combinada com as equações acopladas de Navier-Stokes/convecção-difusão para formular um novo modelo de CG-HDG para resolver problemas de transferência de calor conjugado. Exemplos de referência são resolvidos usando o modelo proposto e validados com valores de literatura.

A parte final da tese aplica a formulação proposta CG-HDG acoplada para prever a resposta térmica de uma secção transversal tubular de GFRP. A equação de radiosidade que governa a radiação interna é adicionada ao modelo acoplado CG-HDG. Os erros de discretização são calculados para estabelecer os intervalos de confiança para quantidades de interesse. Resultados considerando a geometria circular dos cantos da cavidade são apresentados. Estes estão dentro do intervalo de incerteza estimado. Os tempos de CPU requeridos para resolver os sistemas de equações lineares sugerem que o modelo proposto CG-HDG é mais eficiente do que o modelo CG-CG em todos os casos considerados.

palavras-chave: *Hybridizable Discontinuous Galerkin*, acoplamento, problema conjugado de transferência de calor, GFRP, eficiência computacional

Resumen

En esta tesis se propone una formulación acoplada del método de los elementos finitos clásico (CG) y el método *Hybridizable Discontinuous Galerkin* (HDG) para la solución de problemas térmicos conjugados. El modelo se utiliza para determinar la respuesta al fuego de Polímeros Reforzados con Fibras de Vidrio (GFRP) con sección tubular.

El primer paso de la tesis es la comparación de la eficiencia computacional de CG y HDG de alto orden para problemas de flujo incompresible para número de Reynolds (Re) bajo. Se consideran sólo ejemplos 2D y métodos de resolución de sistemas lineales directos. Se presenta una comparación en términos de tiempo de CPU y precisión en la solución para ambas discretizaciones, bajo la misma plataforma de implementación. Los resultados sugieren que HDG puede ser más eficiente computacionalmente que CG en tiempo de CPU, para un grado fijado. La estabilidad de HDG y CG para Re alto se estudia con una solución manufacturada que produce un frente pronunciado, confirmando que HDG proporciona soluciones convergidas suaves en presencia de frentes verticales, en casos en que las oscilaciones numéricas de CG no permiten llegar a convergencia.

A continuación, se plantea la solución del problema acoplado Navier-Stokes/convección-difusión, con la aproximación de Boussinesq, en el contexto del método HDG, y se analiza con experimentos numéricos. Se propone una formulación acoplada HDG-CG para la ecuación del calor. Se comprueban numéricamente las propiedades de convergencia del método propuesto. Finalmente, se combina la formulación acoplada propuesta para la ecuación del calor con el acoplamiento con la ecuaciones de Navier-Stokes en el dominio del fluido, creando una nueva formulación CG-HDG para problemas térmicos conjugados. Se consideran ejemplos clásicos para validar los resultados comparando con la literatura existente.

La parte final de la tesis aplica la formulación acoplada CG-HDG propuesta a la predicción de la respuesta térmica de secciones tubulares de GFRP, incluyendo radiación interna en el modelo. Se calculan estimas de los errores de discretización para determinar intervalos de confianza para las cantidades de interés. Se presentan resultados con geometría con esquinas curvas en la cavidad mostrando resultados dentro de los intervalos de incertidumbre estimados. El tiempo de CPU para la resolución de sistemas sugiere que el modelo CG-HDG propuesto es más eficiente que el clásico método CG-CG en todos los casos considerados.

palabras-clave: *Hybridizable Discontinuous Galerkin*, acoplamiento, transmisión del calor conjugada, GFRP, eficiencia computacional

Acknowledgements

Firstly, I would like to start by expressing my sincere gratitude to my supervisors, Prof. Carlos Tiago and Prof. Sonia Fernández, without whom this work would not have been possible. I had learned a great deal of knowledge from both of them in the past three years and the enormous support and guidance I received from them should get a special mention. Besides teaching me various aspects of classical finite elements and discontinuous Galerkin methods, both of them made me a better researcher by improvising my writing and oral skills.

I also would like to thank Prof. Luís Eça for introducing me to the uncertainty quantification, which helped me to establish the results of GFRP problem. He was so kind to pass by my office on multiple occasions and gave me valuable inputs which helped me to deal with some key issues in this thesis.

I am grateful to Prof. Moitinho de Almeida for organising the project in Lisbon and making sure it ran smoothly over these years. I appreciate his involvement in this work by his advices and suggestions on several problems I had during the course of the project.

Finally, I would like to thank Prof. Pedro Díez, president of Erasmus Mundus SEED project for giving me this excellent opportunity to do my PhD thesis between IST and UPC. I also would like to mention Lelia, secretary of CIMNE who was always available to resolve any bureaucratic issues I had faced along these years.

In the last three years, I had chance meet a lot of amazing people in Lisbon and Barcelona. Everyone was so welcoming and kind to me which helped me to settle down quickly. It is very important to mention the painstaking bicycle rides I had with Carlos which ultimately proved to be fun. The lunch breaks at IST with Prof. Rui Carrilho, Prof. Paulo Teixeira and Prof. Cristina Silva and conspiracy theories about Benfica-Sporting rivalry were memorable. The colleagues from IST and UPC gave me such a pleasant stay during my respective stays. The summer schools gave a very good opportunity to meet the other PhD students and I would like to thank the consortium for organising them. I would like to acknowledge Maria da Conceição for organising my conference travels and summer schools seamlessly.

I would like to express my gratitude to Ana Rosa and Carla Santos of the Área de Pós-Graduação for helping me in the process of submission of my thesis. I would also like to thank Ana Barbosa from IST International Affairs Area for helping me with the paperwork to settle down in Lisbon during my first year.

This work was supported by the Erasmus Mundus Joint Doctorate SEED project financed by European Commission (EACEA). I am grateful to be a part of such a prestigious program and thank everyone who worked to manage the project over all these years.

Finally, I would like to thank my parents and my family who always supported throughout my life which enabled me to reach such a milestone in my life.

Contents

Abstract	iii
Acknowledgements	ix
Contents	xi
List of Figures	xv
List of Tables	xix
List of Symbols	xxi
List of Abbreviations	xxv
1 Introduction	1
1.1 Motivation	1
1.2 Objectives and overview	2
1.3 Discontinuous Galerkin methods	4
1.3.1 Background	4
1.3.2 Interior Penalty method	5
1.3.3 Local discontinuous Galerkin method	6
1.3.4 Compact discontinuous Galerkin method	6
1.4 Hybridizable discontinuous Galerkin method	7
1.5 Coupling of DG and CG methods	11
1.6 Composites thermal models	12
1.7 Notation	12
1.8 Finite element spaces	13
2 Comparison of HDG and CG methods for incompressible Navier–Stokes equations	15
2.1 Overview	15
2.2 CG and HDG discretizations	15
2.2.1 Steady state incompressible Navier–Stokes equations	15
2.2.2 CG formulation	16
2.2.3 HDG formulation	16
2.3 Static condensation	20
2.4 Implementation details	21
2.5 Count of DOFs	22
2.6 Comparison of computational efficiency	25

2.6.1	Kovaszny flow	25
2.6.2	NACA airfoil	31
2.7	Comparison of stability in the presence of sharp fronts	35
2.8	Time discretization using Backward Differentiation Formulae (BDF)	39
2.8.1	Unsteady incompressible Navier–Stokes equations	39
2.8.2	Backward Differentiation Formulae (BDF) schemes	39
2.8.3	Taylor–Vortex flow	42
2.8.4	Error estimation and time step adaption	46
2.8.5	Flow around the cylinder	48
2.9	Conclusions	53
3	Coupling of continuous and hybridizable discontinuous Galerkin methods	55
3.1	Overview	55
3.2	HDG formulation for coupled Navier–Stokes/convection-diffusion equations	55
3.2.1	Governing equations	55
3.2.2	Weak formulation	56
3.2.3	Convergence results	59
3.2.4	Rayleigh–Bénard convection flow	59
3.3	CG-HDG coupled formulation for the heat equation	65
3.3.1	Domain representation	65
3.3.2	Governing equations	66
3.3.3	Strong formulation of the CG-HDG coupled problem	67
3.3.4	Weak formulation of the CG-HDG coupled problem	67
3.3.5	Alternative CG-HDG coupled formulation with a projection operator on the interface	70
3.3.6	Convergence results	72
3.4	Coupled CG-HDG formulation for conjugate heat transfer problem	75
3.4.1	Governing equations	75
3.4.2	Weak formulation	75
3.4.3	Convergence results	76
3.4.4	Conjugate heat transfer problem	81
3.5	Conclusions	85
4	Application to Glass Fiber Reinforced Polymer tubular cross-section	87
4.1	Overview	87
4.2	Problem description	87
4.2.1	Geometry of GFRP cross-section	87
4.2.2	Governing equations	87
4.2.3	Material properties	89
4.3	Radiosity	90
4.3.1	Analytical equation	90
4.3.2	Discretization	91
4.4	Coupled CG-HDG formulation	91
4.4.1	Weak formulation	91
4.4.2	Residual in matrix notation	93
4.4.3	Methodology	93
4.5	Discretization error and uncertainty estimation	95
4.5.1	Numerical errors	95

4.5.2	Discretization error	95
4.5.3	Uncertainty estimation	97
4.6	Numerical results	98
4.6.1	Problem data	98
4.6.2	Development of flow	98
4.6.3	Estimation of uncertainty	103
4.6.4	High-order results	107
4.6.5	Results with rounded corners in cavity	110
4.6.6	Adaptive time step results	111
4.6.7	Experimental validation	117
4.7	Conclusions	119
5	Summary and future developments	121
5.1	Summary	121
5.2	Future work	123
5.2.1	Comparison between CG and HDG	123
5.2.2	Coupled CG-HDG formulation	124
5.2.3	High-order time integration	124
5.2.4	Code development	125
A	Weak formulations for coupled CG-CG and HDG-HDG models	127
B	Theoretical count of degrees of freedoms and number of non-zeros	131
C	Definition of elemental matrices in CG and HDG discretizations	137
	Bibliography	141

List of Figures

1.1	GFRP tubular beam profile considered in this work under experimental setup (Correia <i>et al.</i> , 2010).	1
1.2	Stencils of IPM, LDG and HDG methods for matching triangular mesh. Mesh element under consideration is highlighted in <i>grey</i> and the neighbours that belong to the stencil are represented with <i>yellow</i> . The elements in <i>green</i> do not belong to the stencil. In LDG, only edges of the second neighbours, represented in <i>red</i> , belong to the stencil. In case of HDG, <i>red</i> edges belong to stencil of <i>grey</i> edge and <i>blue</i> edges do not belong to the stencil.	7
1.3	Representation of DOFs for scalar unknown in the final system in CG, DG and HDG methods.	8
2.1	Representation of the nodes in the discretization of the HDG for Navier–Stokes equations.	18
2.2	Comparison of ndof between HDG (T_k/Q_k) and CG ($T_k T_{k-1}/Q_k Q_{k-1}$).	23
2.3	Comparison of ndof between HDG (T_{k-1}/Q_{k-1}) and CG ($T_k T_{k-1}/Q_k Q_{k-1}$).	24
2.4	Sparsity pattern of HDG (T_k) and CG ($T_k T_{k-1}$) matrices for $k = 5$ and $h = 1/2$. Variable nnz represents number of non-zeros.	25
2.5	Kovaszny flow: analytical solution.	26
2.6	Kovaszny flow: convergence of velocity.	27
2.7	Kovaszny flow: convergence of pressure.	27
2.8	Kovaszny flow: error for HDG <i>vs.</i> CPU time for linear solver with triangular elements.	28
2.9	Kovaszny flow: error for CG <i>vs.</i> CPU time for linear solver with triangular elements.	29
2.10	Kovaszny flow: comparison of error between HDG and CG <i>vs.</i> CPU time for linear solver.	29
2.11	Kovaszny flow: ratio of CPU times for linear solver <i>vs.</i> element size for triangular elements.	30
2.12	Kovaszny flow: ratio of CPU times for linear solver <i>vs.</i> element size for quadrilateral elements.	30
2.13	Kovaszny flow: ratio of CPU times for linear solver <i>vs.</i> error for triangular elements.	31
2.14	Kovaszny flow: ratio of CPU times for linear solver <i>vs.</i> error for quadrilateral elements.	32
2.15	Kovaszny flow: Condition numbers in Stokes problem with triangular elements and $h = 1/2^5$	32
2.16	NACA airfoil: computational mesh (refinement level 4) and zoom.	33
2.17	NACA airfoil: velocity field with $k = 5$ and refinement level 4.	34
2.18	NACA airfoil: ratio of CPU times for linear solver <i>vs.</i> element size for triangular elements.	34

2.19	NACA airfoil: ratio of CPU times for linear solver <i>vs.</i> error for triangular elements.	35
2.20	NACA airfoil: condition numbers with triangular elements and $h = 0.78125$	36
2.21	Stability study: isolines of the modulus of the velocity field at $Re = 2000$ and $k = 3$.	37
2.22	Stability study: isolines of the modulus of the velocity field for $Re = 11100$ and $k = 3$	38
2.23	Stability study: relative residual norm <i>vs.</i> iteration number.	38
2.24	Taylor–Vortex flow: analytical solution.	43
2.25	Taylor–Vortex flow: convergence of velocity in time.	43
2.26	Taylor–Vortex flow: convergence of pressure in time.	44
2.27	Taylor–Vortex flow: convergence of velocity in space.	45
2.28	Taylor–Vortex flow: convergence of pressure in space.	45
2.29	Taylor–Vortex flow: convergence of velocity for variable time stepping scheme. . .	46
2.30	Flow around cylinder: computational mesh and zoom.	48
2.31	Flow around cylinder: variation of lift and drag coefficients with time for constant time step of 0.05 sec using HDG and CG.	49
2.32	Flow around cylinder: velocity contours at different time instances.	50
2.33	Flow around cylinder: variation of lift and drag coefficients with time for adaptive time stepping scheme for different levels of local error for CG.	51
2.34	Flow around cylinder: variation of lift and drag coefficients with time for adaptive time stepping scheme for different levels of local error for HDG.	52
2.35	Flow around cylinder: variation of time step with flow time for adaptive time stepping scheme for different levels of local error for CG and HDG discretizations.	52
3.1	Coupled Navier–Stokes/convection-diffusion problem: convergence plots for post-processed velocity and post-processed temperature with HDG using triangular elements.	60
3.2	Coupled Navier–Stokes/convection-diffusion problem: ratio of CPU times for linear solver <i>vs.</i> error in velocity for triangular elements.	60
3.3	Coupled Navier–Stokes/convection-diffusion problem: ratio of CPU times for linear solver <i>vs.</i> error in temperature for triangular elements.	61
3.4	Rayleigh–Bénard convection flow: isolines of post-processed solutions of temperature and velocity at different Ra numbers using HDG.	63
3.5	Rayleigh–Bénard convection flow: distribution of post-processed temperature and velocity components at different Ra numbers using HDG. Circles correspond to reference values in Betts and Haroutunian (1983).	64
3.6	Domain representation: Domain discretization used to present coupled CG-HDG formulation.	66
3.7	Representation of a computational mesh for coupled discretization. <i>Green</i> triangles represent the HDG local elemental variables while the <i>red</i> edges correspond to HDG trace variable. CG mesh is represented in <i>blue</i> and Γ_I is represented in <i>black</i> .	68
3.8	Coupled $CG(T_k)$ -HDG(T_k): convergence plots in Ω	73
3.9	Coupled $CG(T_k)$ -HDG(T_k): convergence plots in Ω_1	73
3.10	Coupled $CG(T_{k+1})$ -HDG(T_k): convergence plots in Ω	74
3.11	Coupled $CG(T_{k+1})$ -HDG(T_k) for conjugate heat transfer problem: convergence plots of velocity and temperature.	77
3.12	Coupled $CG(T_{k+1})$ -HDG(T_k) for conjugate heat transfer problem: ratio of CPU times for linear solver <i>vs.</i> error in temperature for triangular elements. Ratio of CPU times = $CG(T_{k+1})$ -HDG(T_k) time/ $CG(T_{k+1})$ -CG($T_{k+1}T_k$) time.	77

3.13	Coupled CG(T_{k+1})-HDG(T_k) for conjugate heat transfer problem: ratio of CPU times for linear solver <i>vs.</i> error in temperature for triangular elements. Ratio of CPU times = CG(T_{k+1})-HDG(T_k) time/HDG(T_k)-HDG(T_k) time.	79
3.14	Comparison of nnz between coupled CG(T_{k+1})-HDG(T_k) and CG(T_{k+1})-CG($T_{k+1}T_k$) models.	81
3.15	Comparison of nnz between coupled CG(T_{k+1})-HDG(T_k) and HDG(T_k)-HDG(T_k) models.	82
3.16	Conjugate heat transfer problem: geometry and prescribed boundary values of conjugate heat transfer problem domain.	82
3.17	Conjugate heat transfer problem: isolines of temperature and velocity at different Ra numbers using CG-HDG model. Post-processed temperature and velocity are presented in Ω_1	83
4.1	Geometry and boundary conditions notations used for GFRP tubular cross-section All dimensions are in meters.	88
4.2	Thermal properties of GFRP material as a function of temperature.	89
4.3	Density of GFRP material as a function of temperature.	90
4.4	Schematic representation of radiative exchange between surfaces of a cavity. . .	90
4.5	Nested structured meshes used in the analysis of the GFRP problem.	98
4.6	Nested unstructured meshes used in the analysis of the GFRP problem.	99
4.7	GFRP problem: Evolution of temperature and velocity in the cavity at $t = 50, 150, 230$ and 385 sec using coupled CG(Q_2)-HDG(Q_1) model. Post-processed solutions are shown in Ω_f	100
4.7	GFRP problem: Evolution of temperature and velocity in the cavity at $t = 50, 150, 230$ and 385 sec using coupled CG(Q_2)-HDG(Q_1) model. Post-processed solutions are shown in Ω_f . Continuation.	101
4.8	GFRP problem: variation of temperature and Nusselt numbers with time.	102
4.9	Locations of thermocouples in GFRP cross-section.	102
4.10	GFRP problem: evolution of temperature in thermocouples and average Nusselt number with time using mesh M4 and time step used is $0.0625/2^3$. CC and CH denotes CG-HDG and CG-CG models, respectively.	104
4.11	GFRP problem: uncertainty in temperature in thermocouples and average Nusselt number with time using mesh M4 and time step used is $0.0625/2^3$. CC and CH denotes CG-HDG and CG-CG models, respectively.	106
4.12	GFRP problem: evolution temperature in thermocouples and average Nusselt number with time with high-order elements using coupled CG(Q_{k+1})-HDG(Q_k) model. Solid lines and dashed lines correspond to M1 and M2 meshes, respectively and time step used is $0.0625/2^3$	108
4.13	GFRP problem: evolution temperature in thermocouples and average Nusselt number with time with high-order elements using coupled CG(Q_{k+1})-CG($Q_{k+1}Q_k$) model. Solid lines and dashed lines correspond to M1 and M2 meshes, respectively and time step used is $0.0625/2^3$	109
4.14	GFRP problem: evolution temperature in thermocouples and average Nusselt number with time using geometry with rounded corners and coupled CG(T_{k+1})-HDG(T_k) model. Solid lines and dashed lines correspond to UM1 and UM2 meshes, respectively and time step used is $0.0625/2^3$	112

4.15	GFRP problem: evolution temperature in thermocouples and average Nusselt number with time using geometry with rounded corners and coupled $\text{CG}(T_{k+1})$ - $\text{CG}(T_{k+1}Q_k)$ model. Solid lines and dashed lines correspond to UM1 and UM2 meshes, respectively and time step used is $0.0625/2^3$	113
4.16	GFRP problem: evolution temperature in thermocouples and average Nusselt number with time using mesh M4, variable time step and coupled $\text{CG}(Q_2)$ -HDG(Q_1) model.	114
4.17	GFRP problem: evolution temperature in thermocouples and average Nusselt number with time using mesh M4, variable time step and coupled $\text{CG}(Q_2)$ - $\text{CG}(Q_2Q_1)$ model.	115
4.18	GFRP problem: variation of time step with time for the considered coupled models.	116
4.19	GFRP problem: variation of time step and average Nusselt number with time for the considered coupled models.	117
4.20	GFRP problem: experimental data of temperature in thermocouples and numerical results of coupled $\text{CG}(Q_2)$ - $\text{CG}(Q_2Q_1)$ model with time using mesh M4 and time step $0.0625/2^3$ sec.	118
5.1	Barcentric refinement for Scott–Vogelius elements.	123

List of Tables

2.1	Vertices and edges/faces count in terms of number of elements, n_{el}	22
2.2	Analytical expressions for ndof for HDG and CG.	22
2.3	Comparison of DOFs (ndof), number of non-zeros (nnz) in global stiffness matrix (A) and its factor (L) for Stokes problem with $n_{el} = 2048$ triangular elements. .	24
2.4	Maximum Reynolds number for which convergence is achieved, for HDG and CG, for different degree, k	39
2.5	Values of $\beta_{\mu,i}$, $i = 1, \dots, \mu$ and α_{μ} for variable step BDF schemes.	40
2.6	Upper bounds for variable step BDF3 scheme. The scheme is zero-stable if $\omega_{n+1} < R$. .	41
2.7	Partitioning scheme for the time interval $[0, 2]$	44
2.8	Lift, drag coefficients and Strouhal number for $Re = 100$ with constant time step of 0.05 sec.	50
2.9	Time step values in shedding regime for CG and HDG with adaptive time scheme. .	53
2.10	Lift, drag coefficients and Strouhal number for $Re = 100$ with adaptive time stepping scheme.	53
3.1	Rayleigh–Bénard convection flow: stabilization parameters used in the analysis. .	61
3.2	Rayleigh–Bénard convection flow: summary of important quantities and comparison with literature values for $Ra = 10^3 - 10^5$. The velocities of HDG correspond to post-processed solutions.	62
3.3	Rayleigh–Bénard convection flow: summary of important quantities and comparison with literature values for $Ra = 10^6 - 10^8$. The velocities of HDG correspond to post-processed solutions.	65
3.4	Rayleigh–Bénard convection flow: computational details for HDG and CG for $Ra = 10^5$ and $h = 1/2^6$	65
3.5	Expressions of nnz in terms of degree of approximation, k , for the considered coupled models. Symbols n_{el}^f and n_{el}^s corresponds to number of elements in fluid and solid domains, respectively.	80
3.6	Conjugate heat transfer problem: summary of important quantities and comparison with literature values for $Ra = 10^3 - 10^8$	84
3.7	Conjugate heat transfer problem: computational details for CG-HDG and CG-CG for $Ra = 10^5$ and $h = 0.025$	85
4.1	Positions of different thermocouples.	102
4.2	RMS values of errors in temperature for different thermocouples and average Nusselt number for the considered coupled models.	105
4.3	Computational details for coupled CG(Q_2)-CG(Q_2Q_1) and CG(Q_2)-HDG(Q_1) models for GFRP problem with mesh M4 and time step of $0.0625/2^3$	107

4.4	Computational details for coupled $\text{CG}(Q_{k+1})\text{-HDG}(Q_k)$ and $\text{CG}(Q_{k+1})\text{-CG}(Q_{k+1}Q_k)$ models for GFRP problem with mesh M1 ($k = 7$) and mesh M2 ($k = 3$) with a time step of $0.0625/2^3$	110
4.5	Computational details for coupled $\text{CG}(T_{k+1})\text{-HDG}(T_k)$ and $\text{CG}(T_{k+1})\text{-CG}(T_{k+1}T_k)$ models for GFRP problem with mesh UM1 ($k = 7$) and mesh UM2 ($k = 3$) with a time step of $0.0625/2^3$	111
4.6	Number of time steps needed for reaching different time instances for considered coupled models.	116
B.1	Total number of nodes and interior nodes in a i -dimensional entity for simplices and parallelotopes.	133
B.2	Expressions of the total number of connected nodes to a global node of i -dimensional entity in CG discretization.	133
B.3	Expressions of the total number of connected nodes to a global node of i -dimensional entity in HDG discretization.	134
B.4	Expressions to deduce the ndof for coupled models considered in the work. . . .	134
B.5	Expressions for nnz ($\hat{u}_1 \rho$) in terms of k and number of elements, n_{el}	135
B.6	Expressions for nnz for several blocks of unknowns for considered coupled models in the present work.	135

List of Symbols

Sets

\mathcal{C}^s	continuity of order s
\mathcal{H}^k	Sobolev space of order k
\mathcal{L}_p	Lebesgue space of order p
\mathbb{R}^d	space of real numbers of dimension d
\cup	set union
\cap	set intersection
\emptyset	empty set
\in	belongs to
\subset	is a subset of
\forall	for all

Operators

$\delta(\cdot)$	variation of, virtual quantity or iterative quantity
$\text{grad}(\cdot)$	gradient of, differential operator
$\ (\cdot)\ $	norm of
$\text{div}(\cdot)$	divergence of, differential operator
$[\![\cdot]\!]$	jump of
$\max(\cdot)$	maximum of
$\min(\cdot)$	minimum of

Operations

\cdot	dot product of vectors
\otimes	tensorial product of vectors

General quantities

\mathbf{I}	identity second-order tensor
$\mathbf{0}$	zero first-order tensor, zero column vector of arbitrary dimension

Roman letters

b_1, b_2, b_3	constants in adaptive time step scheme
c_h	error constant in space
c_t	error constant in time
c_p	specific heat capacity

c_p^s	specific heat capacity of the GFRP material
c_p^f	specific heat capacity of the air
C_l	lift coefficient
C_d	drag coefficient
d	number of spatial dimensions
D	diameter of cylinder
$e_{n,1}, e_{n,2}$	local error in time step adaption
\mathbf{e}_i	unit vector in orthonormal base co-ordinate system
f_s	shedding frequency
\mathbf{f}	Boussinesq body force vector
F_l	lift force
F_d	drag force
\bar{g}	heat generation
\mathbf{g}	gravity acceleration vector
h	element size
h_a	convective heat transfer coefficient
k	degree of approximation
L	characteristic length
\mathbf{L}	velocity gradient tensor
m_{el}	number of elements in Ω_1
n	number of time steps
n_d	number of observations
n_{el}	number of elements in Ω
n_l	number of elements in Γ_R
n_i	number of i dimensional entities
\mathbf{n}	unit outward normal vector
\overline{Nu}	average Nusselt number
\overline{Nu}_{ma}	moving average of average Nusselt number
p	kinematic pressure
p_{el}	number of elements in Ω_2
Pr	Prandtl number
q	order of convergence in space
\mathbf{q}	heat flux vector
\bar{q}_n	prescribed heat flux
Q_k	quadrilateral element of degree k
Q_{LS}	least-squares functional
r	distance between the points of two surfaces that exchange radiative heat flux in a 3-D space
\mathbf{r}	distance vector
r_h	refinement ratio
R	radiosity
Ra	Rayleigh number
Re	Reynolds number

R_b	upper bound on time step ratios
$\bar{\mathbf{s}}$	prescribed body force vector
S	generic surface
St	Strouhal frequency
t	time instance
$\bar{\mathbf{t}}$	prescribed traction vector
T	total time
T_k	triangular element of degree k
\mathbf{u}	velocity vector
\mathbf{u}^*	post-processed velocity vector
$\bar{\mathbf{u}}$	prescribed velocity vector
$\hat{\mathbf{u}}$	trace velocity vector
U_{exp}	experimental uncertainty
U_{inp}	input uncertainty
U_{val}	validation uncertainty
U_ϕ	uncertainty in ϕ
w	length of the window in moving average method
x_i	spatial co-ordinate

Greek letters

α	thermal diffusivity
α_a	angle of attack
α_μ	coefficient in BDF scheme of order μ
β	thermal expansion coefficient/Boussinesq term
$\beta_{\mu,i}$	coefficient in BDF scheme of order μ
$\gamma^{(e)}$	angle made by line with internal normal of element e
Γ	union of all edges in finite element mesh in Ω
Γ_l, Γ_r	homogeneous Neumann boundaries in GFRP problem
Γ_t, Γ_b	convective and radiative boundaries in GFRP problem
Γ_R	radiosity boundary in GFRP problem
Γ_1	union of all edges in finite element mesh in Ω_1
Γ_2	union of all edges in finite element mesh in Ω_2
Γ_1^D	Dirichlet boundary of Ω_1
Γ_2^D	Dirichlet boundary of Ω_2
Γ_1^N	Neumann boundary of Ω_1
Γ_2^N	Neumann boundary of Ω_2
δ_{tol}	tolerance in adaptive time stepping scheme
Δ_ϕ	range parameter for quantity ϕ
Δt	time step
$\Delta\theta$	characteristic temperature difference
$\bar{\Delta t}$	adapted time step
ϵ	emissivity
ε	discretization error

θ	temperature in Ω
θ_0	reference temperature
$\bar{\theta}$	prescribed temperature
θ	temperature
$\hat{\theta}$	trace temperature
κ	heat conductivity coefficient
$\kappa(\mathbf{A})$	condition number
λ	constant in Kovasznay flow
λ_e	Lagrangian multiplier for element e
$\boldsymbol{\lambda}$	column matrix of Lagrangian multipliers
μ	order of BDF scheme
ν	viscosity
$\boldsymbol{\xi}$	position vector
Π^h	nodal interpolation
ρ	density
ρ_e	mean of pressure in element e
ρ_0	density at reference state
ρ_f	density of fluid in GFRP problem
ρ_s	density of solid in GFRP problem
σ	Stefan–Boltzmann constant
$\hat{\sigma}_\varepsilon$	standard deviation
$\tau, \tau_u, \tau_\theta$	stabilization constants
ϕ	quantity of interest
ψ	approximation function
ω_n, ω_{n+1}	time step ratios
Ω	domain

List of Abbreviations

AMD	Approximate Minimum Degree
BDF	Backward Differentiation Formulae
BDM	Brezzi–Douglas–Marini
CDG	Compact Discontinuous Galerkin
CG	Continuous Galerkin
CPU	Central Processing Unit
DAE	Differential Algebraic Equation
DG	Discontinuous Galerkin
DIRK	Diagonally Implicit Runge–Kutta
DOFs	Degrees of Freedoms
EDG	Embedded Discontinuous Galerkin
FDM	Finite Difference Method
FFT	Fast Fourier Transform
FEM	Finite Element Method
FRP	Fiber Reinforced Polymer
FSI	Fluid Structure Interaction
FVM	Finite Volume Method
GCI	Grid Convergence Index
GFRP	Glass Fiber Reinforced Polymer
GMRES	Generalized Minimal Residual Method
HDG	Hybridizable Discontinuous Galerkin
IIPG	Incomplete Interior Penalty Galerkin
ILES	Implicit Large-Eddy Simulation
IPM	Interior Penalty Method
ISO	International Organization for Standardization
LBB	Ladyzenskaja–Babuška–Brezzi
LDG	Local Discontinuous Galerkin
LDG-H	Local Discontinuous Galerkin-Hybridizable
MDG	Multi-scale Discontinuous Galerkin
NACA	National Advisory Committee for Aeronautics
ndof	number of degrees of freedoms
NIPG	Non-symmetric Interior Penalty Galerkin
nnz	number of non-zeros
PDE	Partial Differential Equation

RK	Runge–Kutta
RMS	Root Mean Square
ROW	Rosenbrock–Wanner
RT	Raviart–Thomas
SIPG	Symmetric Interior Penalty Galerkin
SUPG	Streamline Upwind Petrov–Galerkin
SWE	Shallow Water Equations
VSIMEX	Variable Step Implicit-Explicit Linear Multi-Step Method

Chapter 1

Introduction

1.1 Motivation

Conjugate heat transfer problem describes the variations of temperature in fluid and solid domains in a coupled system. In solids, conduction is the dominant phenomenon, whereas in fluids, convection usually prevails. There are plenty of applications of conjugate heat transfer models (Dorfman, 2009), like designing effective heat exchangers, forced convection regimes, *etc.* The temperature variation in solid is described by the heat conduction equation, whereas the fluid's is described by Navier–Stokes and energy equations. The solution of these problems, however, is far from trivial as convection dominated regimes in fluids may develop sharp fronts and boundary layers. In solids, the solution to the conduction equation is relatively smooth in the absence of strong non-linearities in the material properties, which can depend on temperature. The problem considered in the present context is the transient thermal response of a 2-D Glass Fiber Reinforced Polymer (GFRP) tubular cross-section under fire, which belongs to the class of conjugate heat transfer problems. GFRP is a composite material



Figure 1.1: GFRP tubular beam profile considered in this work under experimental setup (Correia *et al.*, 2010).

which offer several advantages over traditional materials such as high strength, lightweight, low maintenance, durability, *etc.* Owing to their favourable properties, they are used in plenty of applications that include aeronautic, naval, off-shore structures and civil engineering. In spite of numerous advantages, the fire performance of GFRP composites is the major factor that affects their wider usage in engineering structures. Standard Fiber Reinforced Polymer (FRP) materials are flammable, thus, offering inadequate fire reaction. Mouritz *et al.* (2009) summarised the different processes involving the combustion of polymeric composites namely, thermal, chemical, physical and failure processes. Hence, the fire reaction behaviour of the composites is a complex multi-physics problem. In addition, since composites are made of two or more different materials, their thermo-physical properties, like density and specific heat, are temperature dependent (López, 2017). In addition to conduction in the GFRP cross-section and conduction and convection of the air in the tubular cavity, internal radiation should also be taken into account while seeking for the temperature distribution in the GFRP. The combination of different physical phenomena along with several non-linearities in the mathematical model makes the problem of accurate computation of the temperature distribution in the GFRP a challenging one. Once the temperature distribution in the GFRP is obtained, it can be used to perform the mechanical analysis to understand the failure characteristics. Hence, in the present work, the problem of predicting the thermal response of a 2-D GFRP tubular cross-section is considered as it is crucial in analysing its mechanical behaviour.

This type of conjugate heat transfer flows can form boundary layers that might be difficult to resolve depending on the temperature difference in the system. As the temperature difference increases with time, the natural buoyancy of the air develops into a symmetric pattern of flow inside the cavity. After a certain time instance, the flow becomes unstable losing the symmetry before settling into a regime with an oscillatory behaviour. This regime is difficult to predict accurately because of the boundary layers involved in the problem. Therefore, this problem justifies the use of Discontinuous Galerkin (DG) methods, which are known to be more robust than classical Continuous Galerkin (CG) schemes in the presence of sharp fronts. In the present work, a recent class of DG methods, namely Hybridizable Discontinuous Galerkin (HDG) method, is considered to solve for the natural convection of air in a GFRP tubular cross-section. Even though, HDG methods offer various favourable properties as stated in literature, their applicability to real-world problems is still an open question. Hence, this work attempts to address this question by applying HDG to the stated application of GFRP thermal response and comparing the numerical results of HDG model to the conventional CG model.

1.2 Objectives and overview

One of the main objectives of this thesis is to explore the applicability of HDG to multi-physics problems. The main contributions of the thesis are:

1. A comparative study on computational efficiency and accuracy is made between high-order HDG and CG methods for steady incompressible Navier–Stokes equations. This type of efficiency study is available for second-order elliptic operators in the literature and hence, the study is extended to fluid flow problems in the present work. In addition, a study on the robustness of the methods in presence of sharp fronts is presented using a manufactured solution. This study on efficiency and robustness helps to choose the appropriate method depending on the type of practical application. High computa-

tional efficiency is observed using high-order elements for both HDG and CG methods. In terms of stability, HDG method is effective in alleviating numerical oscillations near sharp fronts. The transient term in unsteady Navier–Stokes equations is discretized using Backward Differentiation Formulae (BDF) schemes for both HDG and CG methods. An adaptive time stepping scheme is proposed by estimating the local error using a high-order BDF scheme. This is presented in the Chapter 2 of the thesis. Results on computational efficiency can also be found at Paipuri *et al.* (2017a).

2. The second part concerns with solving the conjugate heat transfer problem using a coupled CG-HDG discretization. In absence of singularities, the solution in the solid domain is often smooth in conjugate heat transfer problems. Hence, in the present work, the solid domain is discretized using CG method, as it was assumed to be best choice to discretize the smooth part of the solution. Consequently, the proposed CG-HDG coupled model uses HDG to discretize the fluid part and CG to solve for the solid domain. Continuity of temperature and heat flux are the transmission conditions need to be satisfied at the interface in this problem. First, HDG formulation presented for Navier–Stokes in Chapter 2 is extended to the coupled Navier–Stokes/convection-diffusion equations. As a second step, a coupled CG-HDG model formulation is introduced for the heat conduction equation and the convergence properties are established using numerical experiments. Finally, the ideas from the first two steps are combined to propose a coupled CG-HDG model for the conjugate heat transfer problem. This coupled CG-HDG model is compared with CG-CG model, where both fluid and solid domains are discretized using CG method and HDG-HDG model, where both domains are solved with HDG discretization. This part is presented in Chapter 3 of the thesis. The proposed HDG-CG coupled formulation for heat problems, and its application to the solution of coupled heat-flow problems can also be found at Paipuri *et al.* (2017b).
3. Finally, the last part of the thesis presents the problem statement and mathematical model of a GFRP cross-section subjected to fire. The radiosity equation that governs the internal radiation is presented along with its discretization details and its inclusion in the governing equations of conjugate heat transfer problem. The problem is solved using CG-HDG and CG-CG coupled models. In the case of practical problems, the estimation of discretization errors and uncertainty is crucial to assess the numerical solution due to unavailability of analytical solutions. The method to estimate the discretization errors and uncertainty used in present work is discussed in detail. Using different meshes in space and different time steps, uncertainty in different quantities of interest are established. These uncertainty estimates are compared for CG-HDG and CG-CG coupled models. The results from high-order approximations and unstructured meshes are shown to be within the estimated uncertainty intervals for the corresponding coupled models. This discussion is provided in Chapter 4.

In order to develop the contributions of this thesis, several tasks are accomplished. They are summarised as follows:

- A fairer comparison of computational efficiency is made by implementing high-order CG and HDG methods for incompressible Navier–Stokes equations on the same platform. Several libraries are shared between the codes, like evaluation of approximation functions, linear solvers, *etc.* Later, the weak formulations for the conjugate heat transfer model are derived for CG and HDG methods by coupling Navier–Stokes/convection-diffusion equations in fluid domain to heat conduction equation in solid part. The weak formulations of the coupled CG-CG and HDG-HDG models are provided in Appendix A.

- Two variants of weak formulations for coupled CG-HDG in the framework of second-order elliptic operators (heat conduction equation, in the present case) are proposed in this work. Both formulations are implemented on the same platform and it is noticed that they produce practically identical results. The derived weak formulations are extended to the governing equations of the conjugate heat transfer model. The results of CG-HDG, CG-CG and HDG-HDG models are compared with the literature ones for a benchmark conjugate heat transfer problem. In addition, the theoretical count of number of non-zero entries in the global tangent matrix is deduced for the considered coupled models and this can be found in Appendix B.
- The weak formulation of unsteady heat conduction equation is derived considering the non-linear material properties of GFRP material along with convective and radiative boundary conditions. Finally, a radiosity model is included in the CG-HDG and CG-CG models to take into account the internal radiation in GFRP tubular cross-section. The corresponding weak formulations and linearisation details are presented in Chapter 4. The derived elemental matrices and the tangent operators of the non-linear contributions are provided in Appendix C.

In the subsequent section, a brief overview of important DG methods is presented. It is followed by a literature review on HDG methods, details on existing CG and DG coupled methods and thermal models of composite materials are provided.

1.3 Discontinuous Galerkin methods

1.3.1 Background

Over the past years, there had been rapid developments in the computing resources which enabled to solve various complex problems in the real world applications. Still, there are plenty of challenges in solving problems which are unsteady, multi-physics, multi-scale, *etc.* Ever since the introduction of numerical techniques like Finite Difference Methods (FDM), Finite Volume Methods (FVM) and Finite Element Methods (FEM) (Courant, 1943), their application to solve engineering problems had a steep growth. There were other well known schemes proposed like Collocation method, Spectral method, *etc.* All the methods differ in two important aspects namely, the way the solution is approximated and the way the approximated solution satisfies the underlying Partial Differential Equation (PDE). In the case of FDM, the solution is approximated as local polynomial and the PDE is satisfied point-wise at mesh points. Even though the method is simple to implement and feasible for high-order, it becomes ill-suited for complex geometries and discontinuities. FVM uses the cell average solution as local approximation and the PDE is satisfied in conservative form. The definition of fluxes at cell interfaces give rise to different variants of FVM and they are well-suited for linear and non-linear conservation laws. FVM can be applied to complicated geometries but extending the method to high-order on generalised unstructured meshes is still an active area of research. Classical FEM which is also known as Continuous Galerkin (CG) method has properties like geometric flexibility and high-order accuracy. However, the PDE is only satisfied on a global sense, which can destroy the locality of the problem. This can pose stability issues in some problems like convection dominated regimes, wave problems, *etc.*

A combination of FEM and FVM by taking the best properties of two worlds, *i.e.* utilizing a space basis and test functions as in case of FEM but satisfying the PDE like FVM, can offer many desirable properties. This lead to the development of Discontinuous Galerkin

(DG) methods. DG method is a class of finite element methods which uses an element-by-element discontinuous approximation and the information passes through the numerical fluxes defined on element boundary. The first DG method was originally proposed to solve the steady state neutron transport problem by Reed and Hill (1973). The first analysis of the method was presented in Lesaint and Raviart (1974) which concluded that the method has optimal convergence rate of $\mathcal{O}(h^{k+1})$ on a Cartesian grid of element size h using a local polynomial approximation of degree k . The extension of the proposed scheme to non-linear scalar conservation law was first attempted by Chavent and Salzano (1982), but the scheme severely limited the time step size to obtain a stable formulation. Since then, DG methods had been applied to many hyperbolic systems like Maxwell's equations (Cockburn *et al.*, 2004), equations of acoustics (Atkins and Shu, 1996), shallow water equations (Engsig-Karup *et al.*, 2006), *etc.* The extension of classical DG approach to second order operators was started by Bassi and Rebay (1997) by rewriting second-order operator as two first-order equations. This led to the introduction of Local Discontinuous Galerkin (LDG) method (Cockburn and Shu, 1998). Another class of DG methods known as Interior Penalty method (IPM) had already been introduced in Arnold (1982) involving second order operator. These developments enabled DG methods to be applied to fluid flow problems notably viscous.

Even though the class of DG methods stated have properties like high-order accuracy, stability and geometric flexibility, they are often criticised for the solution of implicit problems with second order operators due to their high computational cost compared to other discretization methods. The main reasons for the increased cost are the duplication of Degrees of Freedoms (DOFs) at the element interfaces and inability to use static condensation, a numerical technique that allows to express the interior DOFs in terms of boundary DOFs of an element thereby reducing the final DOF count of the problem, especially in high-order computations. In CG method, the DOFs at the element interfaces are shared by both neighbouring elements, whereas in DG methods each element has its own DOFs at interface. Hence, DOFs are repeated at element edges in 2-D and faces in 3-D in the case of DG methods. As the computational domain increases, DG calculation costs can quickly become prohibitive. This effect can be potentially more expensive in the case of unstructured meshes.

HDG methods are designed to address the main drawbacks of most of the DG methods. In HDG, the DOFs are repeated only at the vertices of the element interfaces in 2-D. Hence, HDG methods have fewer DOFs compared to other classes of DG methods. In addition to this, HDG methods are static condensation amenable. In the next section, a brief introduction to several DG methods used in context of fluid flow problems is presented.

1.3.2 Interior Penalty method

The idea of penalty formulation was proposed in the context of enforcing Dirichlet boundary conditions (Nitsche, 1971) by introducing penalty terms on the boundary of the domain in order to penalize the error between true and approximate solutions. Arnold (1982) used this idea to impose the inter-element continuity by introducing consistent penalty terms. The salient feature of this formulation is that there is no need of decomposing high-order PDEs into first-order as in the case of most other DG methods that are used to discretize second-order operators. The different variants in the method are Symmetric Interior Penalty Galerkin (SIPG), Nonsymmetric Interior Penalty Galerkin (NIPG) and Incomplete Interior Penalty Galerkin (IIPG) methods. In SIPG, two stabilisation terms exist namely, a symmetrizing term corresponding to fluxes obtained after integration by-parts and a penalty term which imposes the weak continuity of the numerical solution. The advantage of adding the symmetrizing

term is the resulting linear systems are symmetric and, thus, more efficient to solve. But this formulation is stable only when the penalty parameter is above certain threshold value. On the other hand, in NIPG the symmetrizing term has the opposite sign as the flux term, thereby losing the symmetry of the formulation. Nevertheless, NIPG is stable for any value of penalty. The analysis of methods, including error bounds and choice of penalty terms, can be obtained from Rivière *et al.* (2001) and Georgoulis and Süli (2005). The SIPG has $k + 1$ order of convergence, whereas NIPG method shows k order of convergence in \mathcal{L}_2 norm when a polynomial of degree k is used to represent the solution. In spite of IPM having several positive properties, like optimal convergence and compact support, their applicability to second order operators is less popular. This is probably due to the requirement of fine tuning the penalty parameter, which depends on mesh size and degree of approximation. When a suitable penalty parameter is chosen, this approach can be competitive compared to other DG methods in terms of accuracy.

1.3.3 Local discontinuous Galerkin method

Local discontinuous Galerkin (LDG) is a mixed method, where second-order operators are split into two first-order equations by introducing an auxiliary variable. Standard DG method is applied to each equation by formulating the weak form for each element and then defining the appropriate numerical fluxes. The choice of the definition of numerical fluxes give rise to different formulations, as can be found in Arnold *et al.* (2000). One of the main drawbacks of the LDG method is that it introduces even more unknown variables because of the mixed formulation. Even though, the new additional unknowns can be condensed out of the system at elemental level, the stencils that arise in LDG method are larger and hence, results in less sparser matrices. LDG discretization results not only in coupling of DOFs of one element to couple with DOFs of neighbouring elements, but also to the DOFs of some elements neighbouring the immediate neighbours. The stencil is represented schematically in fig. 1.2. The LDG weak form of each element can be expressed in primal form by using the so-called lifting operators which express the auxiliary variable in term of numerical solution. The primal weak form, that is obtained after introducing the lifting operators, is analogous the IPM with additional terms. Similar to SIPG method, LDG shows optimal convergence with an order of $k + 1$ in \mathcal{L}_2 norm. But, unlike IPM, LDG is less sensitive to the penalty parameter and hence, it can be more robust.

1.3.4 Compact discontinuous Galerkin method

Compact discontinuous Galerkin (CDG) method (Peraire and Persson, 2008) were first introduced to address the shortcomings of LDG method. CDG method has a more compact stencil compared to LDG and hence, the name compact discontinuous Galerkin method. The lifting operators are defined in such a way that the stencil becomes compact in CDG and at the same time retaining the attractive features of LDG. In addition, CDG scheme proved to be slightly more stable than LDG (Peraire and Persson, 2008). The compact formulation produces sparser matrices which in turn have low memory requirements and computational costs. However, there is a slight computational overhead in computing the lifting operators compared to LDG scheme. Even though, *a priori* error estimates require stabilization parameter of order $\mathcal{O}(h^{-1})$ to get optimal convergence, numerical experiments suggests optimal convergence is achieved by using the parameter of order $\mathcal{O}(1)$ (Peraire and Persson, 2008).

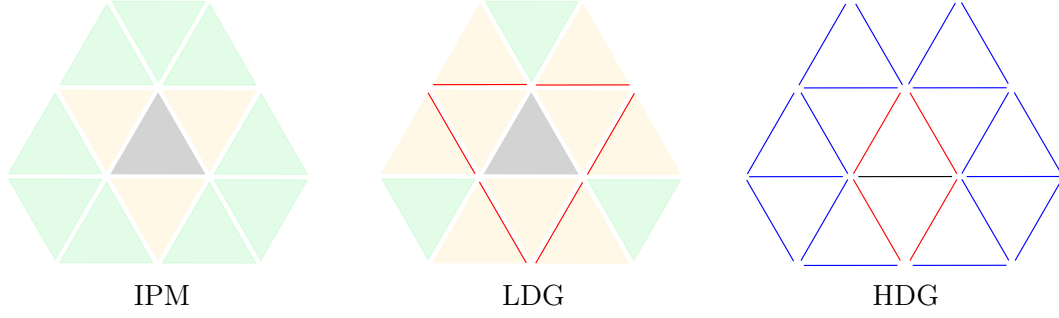


Figure 1.2: Stencils of IPM, LDG and HDG methods for matching triangular mesh. Mesh element under consideration is highlighted in *grey* and the neighbours that belong to the stencil are represented with *yellow*. The elements in *green* do not belong to the stencil. In LDG, only edges of the second neighbours, represented in *red*, belong to the stencil. In case of HDG, *red* edges belong to stencil of *grey* edge and *blue* edges do not belong to the stencil.

1.4 Hybridizable discontinuous Galerkin method

The DG methods discussed earlier have lot of attractive properties like stability, easy adaptivity, able to deal with discontinuities in the solution, *etc.* However, one major drawback that all the previously mentioned methods suffer is high number of DOFs in the final system to be solved. Moreover, numerical techniques like static condensation cannot be applied to those methods as DOFs of one element are dependent on the neighbours. Static condensation can improve the computational efficiency of the discretization scheme at high-order computations. Hence, a static condensation amenable DG discretization would address the serious limitation of other DG schemes and it can be readily parallelized. This resulted in a new class of DG scheme called Hybridizable Discontinuous Galerkin (HDG) method, which was introduced by Cockburn *et al.* (2009b) in the framework of second-order elliptic operator.

Hybridization was introduced in finite element context by Fraeijis de Veubeke (1965). Later, it was shown in Arnold (1985) that the solution of hybridized method can contain more information than the solution of non-hybridized method. Hybridization techniques were first applied in the context of DG methods by Cockburn and Gopalakrishnan (2005), which subsequently lead to development of HDG method. The principle of HDG method is to introduce a new hybrid variable, the so-called trace, on the mesh skeleton. HDG is also a mixed method, like LDG or CDG, where an unknown variable and its derivative are approximated independently. Hence, the final system of unknowns contain a numerical solution of a function, its derivative and the trace variables on mesh skeleton. However, the numerical flux on each interface is defined in such a way that it depends only on the elemental variables and the trace variables shared by the two neighbouring elements in the 2-D case. Therefore, it is possible to condense out the solution and its derivative unknowns element-by-element and express it in terms of the trace. This technique is similar to static condensation in CG method, where interior DOFs are expressed in terms of boundary DOFs inside each element. This is the most attractive feature of HDG method as the final unknowns in the system are just the trace variables, which are defined on mesh skeleton. Also, as stated earlier, the hybrid unknowns have additional information and it is possible to compute a super-convergent solution by a local element-by-element post-processing. Unlike other DG methods, HDG has the optimal rate of convergence of $k + 1$ for solution and its derivatives, whereas the super-convergent solution is of $k + 2$ order, when degree k is used to approximate unknown variables. However,

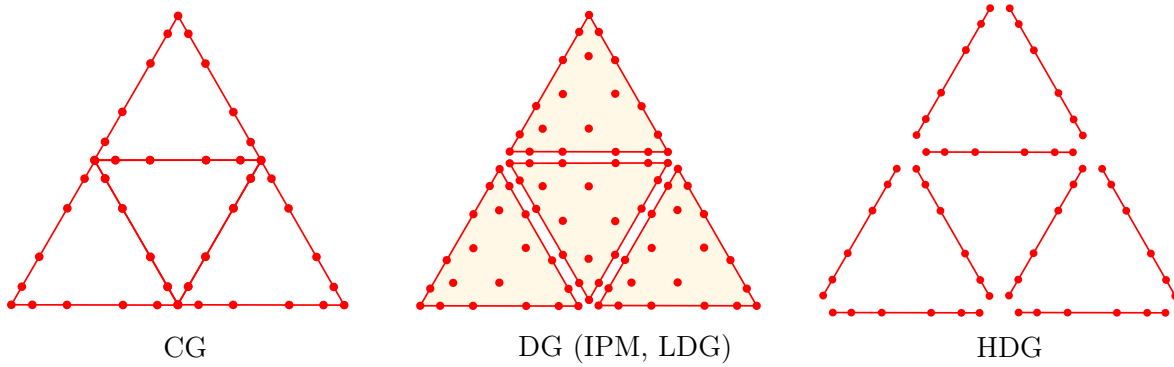


Figure 1.3: Representation of DOFs for scalar unknown in the final system in CG, DG and HDG methods.

the super-convergence in the HDG formulations can only be proved in the diffusion dominated regime. Figure 1.3 shows the fundamental difference between DOFs in CG, DG (IPM, LDG) and HDG methods for the solution of implicit problems with second order operators. In this representation, interior DOFs in CG and local DOFs in HDG are condensed out and only DOFs in the final system of unknowns are represented. The DOFs are repeated at each edge per node in the case of other DG methods, whereas they are repeated only at the vertices of the elements in the case of HDG, when compared to CG counterpart.

After their introduction in the context of second-order operator, HDG methods have been applied to various problems mainly in the case of fluid flow. Cockburn and Gopalakrishnan (2009) applied the HDG method to Stokes problem using the vorticity formulation. They proposed four different hybridization schemes, which differ in the global unknowns on the mesh skeleton. This work mainly deals with the mathematical analysis of different hybridization techniques and did not present any numerical results. Nguyen *et al.* (2010) considered the velocity, gradient of velocity and pressure as local unknowns and trace velocity and mean of pressure as global unknowns for Stokes equations. The numerical tests presented in the paper shows the optimal convergence for all unknowns and super-convergence for velocity. This paper proposed a novel implementation strategy to eliminate the mean of pressure DOFs using an augmented Lagrangian multipliers, thereby leaving only trace velocity DOFs as the global unknowns using an iterative procedure. Cockburn *et al.* (2010) compared different HDG formulations namely, velocity-pressure-gradient, velocity-pressure-stress and velocity-pressure-vorticity for Stokes flow. They did the numerical experiments for the different formulations considered and concluded that velocity-pressure-gradient formulation was the best approximation for the same computational complexity. Therefore, the same formulation is adopted in the present work and used throughout the thesis. They also considered an example with a singularity in the domain and found that all the formulations can only achieve at most order of convergence of 1. Nonetheless, velocity-pressure-gradient formulation produced the lower errors in the case of singularity in the problem. Cockburn *et al.* (2011) considered a new projection based analysis of HDG for Stokes flow and showed that projection of error in velocity super-converges. Cockburn and Shi (2014) presented a review of the recent development of devising HDG methods for Stokes flow. There have been works on variants of Stokes flow using HDG, such as Wang and Khoo (2013), who proposed HDG method with discontinuous viscosity and Gatica and Sequeira (2016), who analysed a augmented HDG method for non-linear Stokes flow.

The application of HDG to Navier–Stokes flow was first reported in Peraire *et al.* (2010).

The report presented the HDG formulation for Euler and Navier–Stokes equations along with numerical examples. The steady state HDG formulations are extended to unsteady cases using backward Euler time integration scheme. Nguyen *et al.* (2011c) considered HDG for Navier–Stokes with velocity-pressure-gradient formulation for both steady and unsteady cases. They used the augmented Lagrangian solution technique developed for Stokes flow in Nguyen *et al.* (2010) to reduce the final unknowns to only trace velocity. Backward Differentiation Formulae (BDF) schemes were used to discretize the time derivative term in this work. Optimal convergence rates were reported for a smooth solution at low Reynolds number and super-convergence was shown for velocity using local post-processing. This paper stated four distinctive features of HDG method for Stokes/Navier–Stokes flows namely, reduced DOFs, optimal convergence, super-convergence and unified treatment of boundary conditions. Moro *et al.* (2011) extended the HDG scheme for Navier–Stokes to turbulence models. In their work, Spalart–Allmaras (SA) turbulence model was used to improve the convergence properties of the method. Giorgiani *et al.* (2014) proposed a degree adaptive algorithm for Navier–Stokes using HDG. In their work, the super-convergent solution was used as error estimator to adapt the degree of approximation in the domain. Several numerical examples in 2-D were analysed and it was concluded that the proposed algorithm could be computationally more efficient than non-adaptive solutions. The analysis of HDG for Navier–Stokes was presented in Cesmelioglu *et al.* (2017). It was proven in the work that by super-penalising the jump of the normal component of the velocity, $\mathcal{H}(\text{div})$ conforming methods are recovered. More recent works include application of HDG to Implicit Large-Eddy Simulation (ILES) flows in Fernandez *et al.* (2017). They used HDG for the discretization of Navier–Stokes equations along with parallel preconditioned Newton-GMRES solver to solve the non-linear system of equations. Numerical tests with Reynolds number as high as 460,000 were presented and a rapid convergence and excellent agreement with experimental data was reported.

HDG were introduced for second-order elliptic operator and they were applied to steady state convection-diffusion-reaction equation in Cockburn *et al.* (2009a). In the paper, they refer the method as Local Discontinuous Galerkin-Hybridizable (LDG-H). In their work, different choices of stabilization parameter and the orders of convergence were provided. They had also compared the proposed HDG method with CG, Brezzi–Douglas–Marini (BDM) (Brezzi *et al.*, 1985) and Raviart–Thomas (RT) (Raviart and Thomas, 1977) methods in terms of convergence rates. They had concluded that a proper choice of stabilization parameter allows the HDG scheme to be very close to classical DG scheme for hyperbolic problems, which proved to be very effective in convective regimes. This work was followed by Nguyen *et al.* (2009b), where they approximated the flux in a different way compared to Cockburn *et al.* (2009a) so that method can deal with purely convective case. The method was extended to time-dependent problems using BDF scheme and numerical tests were presented. It was shown that continuous normal component of flux can be recovered by post-processing the original flux. The work was extended in Nguyen *et al.* (2009a), where they applied HDG for non-linear unsteady convection-diffusion equation. They proposed two different flux formulations in the paper and did the numerical tests to show the optimal convergence rates. An analysis of HDG for convection dominated-diffusion equation was presented in Fu *et al.* (2015b). *A priori* analysis showed that HDG has order of convergence of $k + 1/2$ on general meshes and order of $k + 1$ for the meshes that are aligned in the direction of convection velocity. Nonetheless, numerical experiments confirmed that the optimal convergence of $k + 1$ was obtained even on general meshes in convection-dominated regime. However, the property of super-convergence was lost. They proposed a novel technique to scale the linear system to obtain the spectral condition number independent of the diffusion coefficient. They noticed that the condition

number can be of order $\mathcal{O}(\kappa^{-1})$, where κ is diffusion coefficient and with the proposed scaling, they improved the condition number to $\mathcal{O}(h^{-1})$, where h is the size of the element. Qiu and Shi (2016) proposed a variant of HDG for convection-diffusion equation where polynomials of degree $k + 1$ and k were used to approximate scalar variable and its gradient respectively, where degree k was used to approximate the numerical trace. It was proved that solution of scalar variable converges with order of $k + 2$ in \mathcal{L}_2 norm and this solution was proposed as super-convergent solution as it was two orders higher than the degree of approximation of numerical trace. Other recent works include Cockburn and Mustapha (2015), which applied HDG to fractional diffusion problems and presented the theoretical convergence rates and Wang *et al.* (2016), who proposed HDG to fractional convection-diffusion equation.

Soon *et al.* (2009) applied HDG to linear elasticity problems and an analysis of the method was provided in Fu *et al.* (2015a). They concluded that by using too big stabilization parameter thus, forcing the jump between displacements to be very small, HDG tends to behave as CG method. They also studied the HDG formulation for nearly incompressible elastic materials. They had made the study with a mesh that exhibit locking with CG method and concluded that HDG does not exhibit locking behaviour. It was also stated that optimal convergence was attained at nearly incompressible limit. An example with presence of crack was also provided in their work to demonstrate the HDG ability to handle discontinuities in the solution and its superior stability properties compared to CG method. Celiker *et al.* (2010) proposed HDG for Timoshenko beams and showed the optimal convergence properties using numerical experiments. Nguyen *et al.* (2011a) used HDG formulation for acoustics and Nguyen *et al.* (2011b) applied it for time harmonic Maxwell's equations. Cesmelioglu *et al.* (2013) presented the analysis of HDG method for Oseen (linearised Navier–Stokes) equations. Nguyen and Peraire (2012) provided a brief review of application of HDG method in both fluid and solid mechanics and the related numerical examples. A strategy for solving free surface flows using HDG was introduced in Gürkan *et al.* (2016) using the philosophy of extended finite element methods. Later it was extended to bi-material problems along with Heaviside enrichment in Gürkan *et al.* (2017).

Even though HDG has been successfully applied to different kind of problems, their computational efficiency compared to CG and other discretization techniques is only studied in the context of Laplace equation. There had been a few studies about the efficiency of HDG methods in the past. Most notably, Kirby *et al.* (2012) made a comparative study between HDG and CG for second-order elliptic operator in 2-D. They reported the comparative CPU times for linear solver in case of HDG and CG for the degree of approximation till 14. They had concluded that HDG could be as efficient as CG when degree of five or more was used in the approximation. Later, Yakovlev *et al.* (2016) extended the study to 3-D case, where they made the comparative study between HDG and CG for both direct and iterative solvers. In most of their comparison studies, they considered HDG with degree of approximation k and CG with degree $k + 1$. This is because of the fact that HDG can produce super-convergent solution of order $k + 2$, which can be compared to the CG solution. They found out that HDG can outperform CG using direct solver for degree higher than 3 in case of hexahedral meshes and 5 in case of tetrahedral meshes. In the case of iterative solver, they concluded that HDG was not as efficient as CG and also HDG takes more iterations than CG for obtaining a convergent solution. Moreover, HDG showed better scalability in the case of parallel implementation because of the simplified communication patterns in HDG. In this work, the authors also pointed out the need to develop better pre-conditioners for solving linear systems that arise in HDG. Cockburn *et al.* (2009a) compared HDG and CG methods for convection-diffusion-reaction problem and concluded that HDG of degree k was

as efficient as CG of degree $k + 1$. The work also includes the comparison in terms of \mathcal{L}_2 norm of error in scalar variable and computational complexity which is measure in terms of number of non-zeros (**nnz**) in the stiffness matrix. A comparison of space-time HDG method with space-time DG method was made in Rhebergen *et al.* (2013) and concluded that both approaches perform equally well but reiterated that super-convergent property of HDG can be further exploited. A scalability study of HDG in compressible flows was made in Roca *et al.* (2013) and concluded that GMRES with restart pre-conditioning presents the best weak scaling. A theoretical floating point operations (FLOPS) count for CG, CDG and HDG for second order elliptic problem was presented in Huerta *et al.* (2013). It was concluded that post-processed solution of HDG was as efficient as CG when the operation count, for a given level of accuracy, is considered. A comparison between CG, CDG and HDG for wave problems can be found at Giorgiani *et al.* (2013), where they presented the results of computational efficiency for a given level of accuracy. They stated that HDG exhibit similar performance compared to CG for a given level of accuracy. There have been works on the variants of HDG namely, Embedded Discontinuous Galerkin (EDG) (Güzey *et al.*, 2007) and Multi-scale Discontinuous Galerkin (MDG) (Hughes *et al.*, 2006) methods. These methods have the same number of DOFs as in the case of statically condensed CG system for the same degree of approximation and hence, can be more efficient than HDG methods. But they do not have the super-convergence property which has been one of the stand-out features of HDG method.

Owing to the numerous favourable features of the HDG methods, their application to multi-physics problems is the next logical step after they had been applied to lot of different benchmark problems in both fluid and solid mechanics. Recently, Sheldon *et al.* (2016) proposed a HDG method for modeling fluid-structure interaction (FSI) problems. In their work, HDG formulations for linear and non-linear elastodynamic model and arbitrary Lagrangian-Eulerian Navier–Stokes were derived. A fully coupled monolithic FSI scheme was established in the framework of HDG. Several benchmark examples were solved and concluded that their proposed FSI formulation is well within the results of benchmark results. Prada (2016) applied HDG in the context of non-linear porous media visco-elastic problems applied to ophthalmology. In his work, he solved the equations of viscous fluid coupled with visco-elastic solid and solved resulting the non-linear equations using the fixed point iteration scheme. The work highlighted the fact that the dual variables converge at the same rate as primal variables in HDG which is particularly interesting in their application as dual variables (stresses and discharge velocity) is of engineering importance. The problem of locking is also suppressed using HDG which is common in numerical simulations of poro-elasticity using CG methods. In both the works stated above, HDG was used to discretize all the equations governing the multi-physics problem. This is due to the limitations of discretization techniques like CG in presence of strong discontinuities, locking, *etc.*

1.5 Coupling of DG and CG methods

The coupling of CG and DG methods were first presented by Alotto *et al.* (2001) in the simulation of rotating electrical machines. Later, Perugia and Schötzau (2001) developed the analysis of the coupling for the case of second-order elliptic problems. Both the works were motivated by the ability of DG methods to handle the hanging nodes on non-matching interfaces. Later, the coupling of LDG and CG methods was proposed by Dawson and Proft (2002a) for transport problems. Their motivation was to discretize the part of solution with presence of high gradients with LDG and the smoother part of solution with CG. They demonstrated the advantages of coupling with numerical examples and reported optimal convergence

rates for 1-D problems. Devloo *et al.* (2007) used the formulation presented in Baumann and Oden (1999) for the DG part of solution. Their work considers Streamline Upwind Petrov Galerkin (SUPG) stabilization for convective term of DG discretization and concluded that it enhances the quality of solution.

Coupling of CG and DG methods were also proposed in the context of shallow water equations (SWE) in different variants by Dawson and Proft (2002b, 2003, 2004), Blain and Massey (2005). In those works, DG and CG methods were applied on different governing equations in all the domain. For instance, in the work of Dawson and Proft (2002b), NIPG method was used to discretize the hyperbolic continuity equations of SWE while CG method was used for the momentum equation. A coupled LDG and CG model was studied by Zhu *et al.* (2011b) for 1-D convection-diffusion equation and later extended the model to perturbed problems of convection-diffusion type in Zhu *et al.* (2011a) and Zarin (2009). A coupled formulation of mixed finite element method with DG was proposed by Guo *et al.* (2014) to study the miscible displacement problem in porous media. Liu *et al.* (2009) proposed a coupled continuous-discontinuous Galerkin framework to enhance the applicability and efficiency of DG method to solve large-scale poroelasticity problems. A study on stability of CG-DG Galerkin methods for advection-diffusion-reaction problems was presented in Cangiani *et al.* (2013). In their work, a different coupling strategy without using transmission conditions on interface boundary was proposed. Also, in their earlier work, Cangiani *et al.* (2006) compared the coupled DG-CG model for convection-diffusion equation with DG and stabilised CG methods. As far as the knowledge of the author is concerned, coupled CG-HDG scheme has not been proposed yet in any framework and it is developed in this work in Chapter 3.

1.6 Composites thermal models

The thermal models proposed in literature mainly differs in the type of process being modelled. One of the first models was proposed in Henderson *et al.* (1985), where the thermochemical models developed for wood were applied for composite laminates. Several other works were presented in Florio *et al.* (1991), Miller and Weaver (2003), Miano and Gibson (2009). A finite element model was presented in Sullivan and Salamon (1992a,b) to simulate the thermochemical decomposition of polymeric material. The thermochemical response was used to perform the mechanical analysis and the model was validated with experimental data. A transient non-linear finite element model was presented in Looyeh and Bettess (1998) by considering the temperature-dependent thermal properties of GFRP and different mixed boundary conditions. A similar thermochemical study was made using finite element simulation by Keller *et al.* (2006) for liquid-cooled and non-cooled slab components. In their work, the authors mentioned that it was important to consider the natural convection of air and internal radiation in enclosed spaces of composite sections. Recently, López (2017) did a coupled multi-physics finite element simulation of GFRP tubular cross section considering natural convection of air and internal radiation inside the enclosed cavity. In that thesis, the author states that natural convection of the air can be a bottleneck in the solution process.

1.7 Notation

Most of the algebra presented in this text is expressed in symbolic (also frequently referred to as direct, intrinsic or absolute) notation (Holzapfel, 2000). The usual matrix and indicial notation are sometimes employed in specific cases.

Throughout the text italic Latin or Greek lowercase letters ($a, b, \dots, \alpha, \beta, \dots$) denote scalar quantities, bold italic Latin or Greek lowercase letters ($\mathbf{a}, \mathbf{b}, \dots, \boldsymbol{\alpha}, \boldsymbol{\beta}, \dots$) denote vectors and bold italic Latin or Greek capital letters ($\mathbf{A}, \mathbf{B}, \dots$) denote second-order tensors in a d -dimensional Euclidean space.

Rectangular and single-column matrices built of tensor components on orthogonal frames are expressed by boldface upright Latin or Greek letters ($\mathbf{A}, \mathbf{B}, \dots, \mathbf{a}, \mathbf{b}, \dots, \boldsymbol{\rho}, \boldsymbol{\lambda}, \dots$). The scalar products used in the present paper are $(\cdot, \cdot)_D$ and $\langle \cdot, \cdot \rangle_B$, which represents the \mathcal{L}_2 scalar product in any domain, D , and over any boundary, B , respectively.

Let \mathbf{w} and z be generic vector and scalar fields, respectively, defined over Ω . Their error norms are computed as follows,

$$\begin{aligned} \|e_{\mathbf{w}}\|_{\mathcal{L}_2(\Omega)} &= \left[\int_{\Omega} (\mathbf{w}_{ex} - \mathbf{w}_{num}) \cdot (\mathbf{w}_{ex} - \mathbf{w}_{num}) d\Omega \right]^{1/2}, \\ \|e_z\|_{\mathcal{L}_2(\Omega)} &= \left[\int_{\Omega} (z_{ex} - z_{num})^2 d\Omega \right]^{1/2}, \end{aligned} \quad (1.1)$$

where suffixes ex and num stand for exact and numerical values. Throughout this thesis, triangular and quadrilateral elements are designated by T_k and Q_k , respectively, where k is the degree of approximation.

1.8 Finite element spaces

The finite element spaces used in the present work are defined in this section. All the problems considered are two-dimensional. If Ω and Γ are generic domain and boundary, respectively, different finite element spaces are defined as follows,

$$\begin{aligned} \mathcal{V}_k^h(\Omega) &:= \{v \in \mathcal{L}_2(\Omega) : v|_{\Omega^e} \in \mathcal{P}_k(\Omega^e), \forall \Omega^e \subset \Omega\}, \\ \Lambda_k^h(\Gamma) &:= \{\hat{v} \in \mathcal{L}_2(\Gamma) : \hat{v}|_{\Gamma^i} \in \mathcal{S}_k(\Gamma^i), \forall \Gamma^i \subset \Gamma\}, \\ \mathcal{W}_k^h(\Omega) &:= \{w \in \mathcal{H}^1(\Omega) : w|_{\Omega^e} \in \mathcal{P}_k(\Omega^e), \forall \Omega^e \subset \Omega\}, \\ \Sigma_k^h(\Gamma) &:= \{z \in \mathcal{L}_2(\Gamma) : z|_{\Gamma^i} \in \mathcal{S}_k(\Gamma^i), \forall \Gamma^i \subset \Gamma\}, \end{aligned} \quad (1.2)$$

where subscript k denotes the degree of polynomial and \mathcal{P}_k is the space of polynomials of degree less than or equal to k . For instance, $\mathcal{W}_{k+1}^h(\Omega)$ spans the polynomials of degree less than or equal to $k+1$. The standard segment, \mathcal{S}_k , triangular, \mathcal{T}_k , and quadrilateral, \mathcal{Q}_k regions are defined as,

$$\begin{aligned} \mathcal{S}_k(\Gamma^i) &:= \{s^p ; 0 \leq p \leq k ; (x_1(s), x_2(s)) \in \Gamma^i ; -1 \leq s \leq 1\}, \\ \mathcal{T}_k(\Omega^e) &:= \{\xi_1^p \xi_2^q ; 0 \leq p+q \leq k ; (x_1(\xi_1, \xi_2), x_2(\xi_1, \xi_2)) \in \Omega^e ; -1 \leq \xi_1, \xi_2, \xi_1 + \xi_2 \leq 0\}, \\ \mathcal{Q}_k(\Omega^e) &:= \{\xi_1^p \xi_2^q ; 0 \leq p, q \leq k ; (x_1(\xi_1, \xi_2), x_2(\xi_1, \xi_2)) \in \Omega^e ; -1 \leq \xi_1, \xi_2 \leq 1\}. \end{aligned} \quad (1.3)$$

Depending on the type of elements used, \mathcal{P}_k space in (1.2) is computed in one of the standard elements, *i.e.*, \mathcal{T}_k or \mathcal{Q}_k defined in (1.3).

Chapter 2

Comparison of hybridizable discontinuous Galerkin and continuous Galerkin methods for incompressible Navier–Stokes equations

2.1 Overview

A comparison of computational efficiency for the velocity-gradient-pressure formulation of HDG and velocity-pressure CG formulation with Taylor–Hood elements is presented in this chapter. The results are provided in terms of DOF count, number of non-zeros and CPU times. Later, time discretization of unsteady Navier–Stokes equations is presented using Backward Differentiation Formulae (BDF) schemes. An adaptive time stepping scheme is presented and verified using the example of laminar flow around cylinder.

2.2 The CG and HDG discretization of the incompressible Navier–Stokes equations

2.2.1 Steady state incompressible Navier–Stokes equations

Let Ω be the domain with boundary $\partial\Omega$ divided into Dirichlet, $\partial\Omega_D$, and Neumann, $\partial\Omega_N$, boundaries. The steady state incompressible Navier–Stokes equations can be written as

$$\operatorname{div}(\mathbf{u} \otimes \mathbf{u}) - \operatorname{div}(-p\mathbf{I} + \nu \operatorname{grad} \mathbf{u}) = \bar{\mathbf{s}} \quad \text{in } \Omega, \quad (2.1a)$$

$$\operatorname{div} \mathbf{u} = 0 \quad \text{in } \Omega, \quad (2.1b)$$

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } \partial\Omega_D, \quad (2.1c)$$

$$(-p\mathbf{I} + \nu \operatorname{grad} \mathbf{u}) \mathbf{n} = \bar{\mathbf{t}} \quad \text{on } \partial\Omega_N, \quad (2.1d)$$

where \mathbf{u} is the velocity, p is the kinematic pressure, ν is the kinematic viscosity, $\bar{\mathbf{s}}$ is the body force, $\bar{\mathbf{u}}$ is the prescribed velocity on the Dirichlet boundary, $\partial\Omega_D$, and $\bar{\mathbf{t}}$ is the prescribed pseudo traction on the Neumann boundary, $\partial\Omega_N$.

The next subsections present the basics on the CG and the HDG discretizations considered in this work.

2.2.2 CG formulation

The CG weak form of the equilibrium equations, (2.1a) and (2.1d), and the incompressibility condition, (2.1b), can be stated as,

$$\begin{aligned} (\delta \mathbf{u}, (\operatorname{div}(\mathbf{u} \otimes \mathbf{u}) - \operatorname{div}(-p\mathbf{I} + \nu \operatorname{grad} \mathbf{u}) - \bar{\mathbf{s}}))_{\Omega} + \langle \delta \mathbf{u}, ((-p\mathbf{I} + \nu \operatorname{grad} \mathbf{u}) \mathbf{n} - \bar{\mathbf{t}}) \rangle_{\partial\Omega_N} &= 0, \\ -(\delta p, \operatorname{div} \mathbf{u})_{\Omega} &= 0. \end{aligned} \quad (2.2)$$

Using the divergence theorem, integration by-parts, identity $\operatorname{div}(\mathbf{u} \otimes \mathbf{u}) = \mathbf{u}(\operatorname{div} \mathbf{u}) + (\operatorname{grad} \mathbf{u}) \mathbf{u}$ and setting $\operatorname{div} \mathbf{u} = 0$, CG weak form is: find $\mathbf{u} \in [\mathcal{H}^1(\Omega)]^2$ and $p \in \mathcal{L}_2(\Omega)$ such that $\mathbf{u} = \Pi^h(\bar{\mathbf{u}})$ on $\partial\Omega_D$ and

$$\begin{aligned} (\delta \mathbf{u}, (\operatorname{grad} \mathbf{u}) \mathbf{u})_{\Omega} + (\operatorname{grad} \delta \mathbf{u}, -p\mathbf{I} + \nu \operatorname{grad} \mathbf{u})_{\Omega} - (\delta \mathbf{u}, \bar{\mathbf{s}})_{\Omega} - \langle \delta \mathbf{u}, \bar{\mathbf{t}} \rangle_{\partial\Omega_N} &= 0, \\ -(\delta p, \operatorname{div} \mathbf{u})_{\Omega} &= 0, \end{aligned} \quad (2.3)$$

for all $\delta \mathbf{u} \in [\mathcal{H}^1(\Omega)]^2$, such that $\delta \mathbf{u} = \mathbf{0}$ on $\partial\Omega_D$, and for all $\delta p \in \mathcal{L}_2(\Omega)$.

If the problem is a pure Dirichlet one, that is $\partial\Omega = \partial\Omega_D$ and $\partial\Omega_N = \emptyset$, the pressure is determined up to a constant. In this case, the mean of the pressure in Ω is set to a prescribed value to ensure uniqueness of the solution.

The weak form is discretized with mixed Taylor–Hood approximations (Taylor and Hood, 1973), with degree k for the velocities, \mathbf{u} and $\delta \mathbf{u}$, and degree $k-1$ for the pressures, p and δp , satisfying the so-called LBB condition (Brezzi and Fortin, 1991) for stability. The residual of the Navier–Stokes equations, after spatial discretization, can be expressed as follows,

$$\mathbf{r}(\mathbf{u}, \mathbf{p}) \equiv \begin{bmatrix} \mathbf{K} + \mathbf{C}(\mathbf{u}) & \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{p} \end{Bmatrix} - \begin{Bmatrix} \bar{\mathbf{s}} + \bar{\mathbf{t}} \\ \mathbf{0} \end{Bmatrix} = \mathbf{0}. \quad (2.4)$$

In the equation (2.4), \mathbf{K} , \mathbf{G} and $\mathbf{C}(\mathbf{u})$ represent the viscosity, discrete gradient operator and convective matrices, respectively, while $\bar{\mathbf{s}}$ and $\bar{\mathbf{t}}$ contains the body force and traction vectors, respectively. The nodal values of velocity and pressure are represented by \mathbf{u} and \mathbf{p} , respectively. The elemental matrices used to compute the global system are presented in Appendix C.

The CG formulation, with Taylor–Hood approximations of degree k for velocity and $k-1$ for pressure, leads to errors of order k in \mathcal{H}^1 norm for velocity and in \mathcal{L}_2 norm for pressure and, consequently, errors of order $k+1$ in \mathcal{L}_2 norm are expected for the velocity solution.

This work does not consider stabilized formulations, which would be necessary in highly convection-dominated problems. Computational efficiency is studied with numerical examples that do not present sharp fronts, aiming for a fair comparison of the accuracy and CPU time.

2.2.3 HDG formulation

Again, let Ω be the domain with boundary $\partial\Omega$ divided into Dirichlet, $\partial\Omega_D$, and Neumann, $\partial\Omega_N$, boundaries. Assuming Ω is divided into n_{el} number of non-overlapping elements, Ω^e , the union of all interior edges in the finite element mesh is denoted by Γ , i.e., $\Gamma = \bigcup_{e=1}^{n_{el}} \partial\Omega^e$.

HDG is a mixed formulation method which requires to rewrite the second-order equations into two first order equations. The Navier–Stokes equation inside each element can be rewritten as follows by taking the account of discontinuities in the approximation spaces between

elements,

$$\begin{aligned} \mathbf{L} - \text{grad } \mathbf{u} &= \mathbf{0} \quad \text{in } \Omega^e, \\ \text{div } (\mathbf{u} \otimes \mathbf{u}) - \text{div } (-p\mathbf{I} + \nu\mathbf{L}) &= \bar{\mathbf{s}} \quad \text{in } \Omega^e, \end{aligned} \quad (2.5a)$$

$$\text{div } \mathbf{u} = 0 \quad \text{in } \Omega^e,$$

$$[\![\mathbf{u} \otimes \mathbf{n}]\!] = \mathbf{0} \quad \text{on } \Gamma \setminus \partial\Omega, \quad (2.5b)$$

$$[\![-p\mathbf{I} + \nu\mathbf{L}]\!] \mathbf{n} = \mathbf{0} \quad \text{on } \Gamma \setminus \partial\Omega, \quad (2.5c)$$

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } \partial\Omega_D, \quad (2.5d)$$

$$(-p\mathbf{I} + \nu\mathbf{L}) \mathbf{n} = \bar{\mathbf{t}} \quad \text{on } \partial\Omega_N,$$

where the new variable \mathbf{L} is the gradient of velocity and equations (2.5b) and (2.5c) impose the continuity of velocity and normal component of pseudo-stress across the element boundaries, respectively. The jump operator, $[\![\cdot]\!]$, is introduced following the definition by Montlaur *et al.* (2012). It is defined along the mesh skeleton, Γ , and it involves values from the elements to the left and the right of the trace, Ω^i and Ω^j ,

$$[\![\odot]\!] = \odot_i + \odot_j, \quad (2.6)$$

The definition of the jump operator always involves the normal to the edge of the element.

The salient feature of HDG is the introduction of an independent approximation for the trace of the velocity, $\hat{\mathbf{u}}$, on the mesh skeleton, Γ . The introduction of this trace velocity defines two types of problems, namely local and global. The local problem corresponds to the solution of Navier–Stokes equations inside each element with $\hat{\mathbf{u}}$ as Dirichlet boundary condition. However, Navier–Stokes problem with Dirichlet boundary condition on all the boundary is not solvable unless a condition on pressure is imposed. Hence, a new variable, $\rho_e \in \mathbb{R}^{n_{el}}$, is introduced as the mean of the pressure on the boundary of each element. The strong form of local problem can be written as,

$$\begin{aligned} \mathbf{L} - \text{grad } \mathbf{u} &= \mathbf{0} \quad \text{in } \Omega^e, \\ \text{div } (\mathbf{u} \otimes \mathbf{u}) - \text{div } (-p\mathbf{I} + \nu\mathbf{L}) &= \bar{\mathbf{s}} \quad \text{in } \Omega^e \\ \text{div } \mathbf{u} &= 0 \quad \text{in } \Omega^e, \end{aligned} \quad (2.7a)$$

$$\mathbf{u} = \hat{\mathbf{u}} \quad \text{on } \partial\Omega^e, \quad (2.7b)$$

$$\frac{1}{|\partial\Omega^e|} \langle p, 1 \rangle_{\partial\Omega^e} = \rho_e, \quad (2.7c)$$

for $e = 1, \dots, n_{el}$. Figure 2.1a shows a representation of local variables and global variables in blue and red, respectively. The mean of the pressure, ρ_e , is a scalar value defined for each element, Ω^e and it is denoted by green dot in fig. 2.1a. The weak form of the local problem using appropriate weighting functions can be written as follows,

$$(\delta\mathbf{L}, (\mathbf{L} - \text{grad } \mathbf{u}))_{\Omega^e} + \langle (\delta\mathbf{L}\mathbf{n}, (\mathbf{u} - \hat{\mathbf{u}})) \rangle_{\partial\Omega^e} = 0, \quad (2.8a)$$

$$\begin{aligned} (\delta\mathbf{u}, \text{div } (\mathbf{u} \otimes \mathbf{u}))_{\Omega^e} - (\delta\mathbf{u}, \text{div } (-p\mathbf{I} + \nu\mathbf{L}))_{\Omega^e} + \langle \delta\mathbf{u}, \tau_u (\mathbf{u} - \hat{\mathbf{u}}) \rangle_{\partial\Omega^e} \\ + \langle \delta\mathbf{u}, (\hat{\mathbf{u}} \otimes \hat{\mathbf{u}} - \mathbf{u} \otimes \mathbf{u}) \mathbf{n} \rangle_{\partial\Omega^e} - (\delta\mathbf{u}, \bar{\mathbf{s}})_{\Omega^e} = 0, \end{aligned} \quad (2.8b)$$

$$(\delta p, \text{div } \mathbf{u})_{\Omega^e} + \langle \delta p, (\mathbf{u} - \hat{\mathbf{u}}) \cdot \mathbf{n} \rangle_{\partial\Omega^e} = 0, \quad (2.8c)$$

where the Dirichlet boundary condition (2.7b) is applied weakly on all the equations and the stabilization constant, τ_u , is introduced in equation (2.8b) to ensure dimensional consistency.

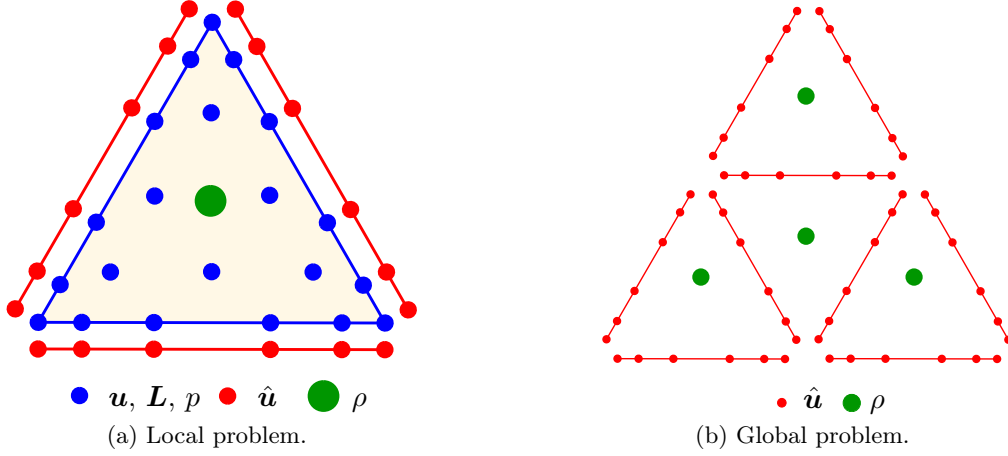


Figure 2.1: Representation of the nodes in the discretization of the HDG for Navier–Stokes equations.

After using integration by-parts, the problem becomes: find $(\mathbf{u}, p, \mathbf{L}, \hat{\mathbf{u}}, \rho_e) \in [\mathcal{V}_k^h(\Omega)]^2 \times [\mathcal{V}_k^h(\Omega)] \times [\mathcal{V}_k^h(\Omega)]^{2 \times 2} \times [\Lambda_k^h(\Gamma)]^2 \times \mathbb{R}^{n_{el}}$ satisfying the *local problem* in every element Ω^e ,

$$(\delta \mathbf{L}, \mathbf{L})_{\Omega^e} + (\operatorname{div} \delta \mathbf{L}, \mathbf{u})_{\Omega^e} - \langle \delta \mathbf{L} \mathbf{n}, \hat{\mathbf{u}} \rangle_{\partial \Omega^e} = 0, \quad (2.9a)$$

$$- (\operatorname{grad} \delta \mathbf{u}, \mathbf{u} \otimes \mathbf{u})_{\Omega^e} - (\delta \mathbf{u}, \operatorname{div} (-p \mathbf{I} + \nu \mathbf{L}))_{\Omega^e} + \langle \delta \mathbf{u}, (\hat{\mathbf{u}} \otimes \hat{\mathbf{u}}) \mathbf{n} + \tau_u (\mathbf{u} - \hat{\mathbf{u}}) \rangle_{\partial \Omega^e} - (\delta \mathbf{u}, \bar{\mathbf{s}})_{\Omega^e} = 0, \quad (2.9b)$$

$$- (\operatorname{grad} \delta p, \mathbf{u})_{\Omega^e} + \langle \delta p, \hat{\mathbf{u}} \cdot \mathbf{n} \rangle_{\partial \Omega^e} = 0, \quad (2.9c)$$

$$\frac{1}{|\partial \Omega^e|} \langle p, 1 \rangle_{\partial \Omega^e} = \rho_e, \quad (2.9d)$$

for all $(\delta \mathbf{u}, \delta p, \delta \mathbf{L}, \delta \hat{\mathbf{u}}) \in [\mathcal{V}_k^h(\Omega)]^2 \times [\mathcal{V}_k^h(\Omega)] \times [\mathcal{V}_k^h(\Omega)]^{2 \times 2} \times [\Lambda_k^h(\Gamma)]^2$ and $e = 1, \dots, n_{el}$. The discrete spaces are defined in (1.2). The equation (2.9b) can be rewritten using integration by-parts as follows,

$$- (\operatorname{grad} \delta \mathbf{u}, \mathbf{u} \otimes \mathbf{u})_{\Omega^e} + (\operatorname{grad} \delta \mathbf{u}, (-p \mathbf{I} + \nu \mathbf{L}))_{\Omega^e} + \langle \delta \mathbf{u}, (\hat{\mathbf{u}} \otimes \hat{\mathbf{u}}) \mathbf{n} \rangle_{\partial \Omega^e} - \langle \delta \mathbf{u}, (-p \mathbf{I} + \nu \mathbf{L}) \mathbf{n} + \tau_u (\hat{\mathbf{u}} - \mathbf{u}) \rangle_{\partial \Omega^e} - (\delta \mathbf{u}, \bar{\mathbf{s}})_{\Omega^e} = 0. \quad (2.10)$$

Using equation (2.10), it is possible to define the so-called numerical trace flux of the HDG on the each element boundary as follows,

$$(-\widehat{p \mathbf{I} + \nu \mathbf{L}}) \mathbf{n} = (-p \mathbf{I} + \nu \mathbf{L}) \mathbf{n} + \tau_u (\hat{\mathbf{u}} - \mathbf{u}). \quad (2.11)$$

The local problem (2.9) for each element allows expressing the velocity, \mathbf{u} , the gradient of the velocity, \mathbf{L} , and the pressure, p , in terms of the trace of the velocity, $\hat{\mathbf{u}}$, on the mesh skeleton and the mean of the pressure at the element, ρ_e . Therefore, $\hat{\mathbf{u}}$ and ρ_e can be regarded as actual unknowns of the problem, that are determined with the global problem.

Equations (2.5b) and (2.5c) define the global problem. The trace of velocity is defined to be single-valued for the neighbouring elements and hence, jump is zero which satisfies equation (2.5b). Equation (2.5c) corresponds to so-called conservativity condition, *i.e.*, the conservation of the normal fluxes across interior faces of the mesh skeleton. The global problem is defined using the conservativity condition (2.11) across interior faces, together

with the Neumann and Dirichlet boundary conditions, and the solvability condition (2.12b) for the Dirichlet data in the local problems. The *global problem* is stated as follows,

$$\sum_{e=1}^{n_{el}} \langle \delta \hat{\mathbf{u}}, (-p\mathbf{I} + \nu \mathbf{L}) \mathbf{n} + \tau_u (\hat{\mathbf{u}} - \mathbf{u}) \rangle_{\partial \Omega^e} = \langle \delta \hat{\mathbf{u}}, \bar{\mathbf{t}} \rangle_{\partial \Omega_N}, \quad (2.12a)$$

$$\langle \hat{\mathbf{u}} \cdot \mathbf{n}, 1 \rangle_{\partial \Omega^e} = 0 \quad \text{for } e = 1, \dots, n_{el}, \quad (2.12b)$$

$$\hat{\mathbf{u}} = \mathbb{P}_2(\bar{\mathbf{u}}) \quad \text{on } \partial \Omega_D, \quad (2.12c)$$

for all $(\delta \mathbf{u}, \delta p, \delta \mathbf{L}, \delta \hat{\mathbf{u}}) \in [\mathcal{V}_k^h(\Omega)]^2 \times [\mathcal{V}_k^h(\Omega)] \times [\mathcal{V}_k^h(\Omega)]^{2 \times 2} \times [\Lambda_k^h(\Gamma)]^2$, such that $\delta \hat{\mathbf{u}} = \mathbf{0}$ on $\partial \Omega_D$, where the discrete spaces are defined in (1.2) and $\mathbb{P}_2(\bar{\mathbf{u}})$ is the \mathcal{L}_2 projection of the Dirichlet data into the approximation space on $\partial \Omega_D$. In the case of a pure Dirichlet problem, that is $\partial \Omega = \partial \Omega_D$, the mean of the pressure ρ_e is set to a constant in, for instance, a single element, closing the problem with an unique solution. Following Nguyen *et al.* (2011c), τ_u is a positive parameter, and it is usually taken as

$$\tau_u \approx \frac{\nu}{L} + |\mathbf{u}|, \quad (2.13)$$

where L is the characteristic length of the problem. Based on dimensional analysis, τ_u has same dimensions as the ratio ν/L and the velocity magnitude. The former accounts for the diffusion effect while the later for the convection effect. Even though the so-called stabilization parameter has some influence on the accuracy of the HDG solution, the method is very robust to variations of τ_u (Cockburn *et al.*, 2011). Nevertheless, as will be seen in the numerical tests in Section 2.7, this parameter may have an important effect on the stability properties of the HDG method, to alleviate or remove numerical oscillations in the presence of sharp fronts. In the present work, for sake of simplicity, τ_u is either taken as 1 or in the case of high-convective flows, it is taken as the expected magnitude of maximum velocity in the whole domain.

The discretization of local and global problems (2.9) and (2.12) leads to a discrete residual of the form

$$\mathbf{r} = \left[\begin{array}{cc|ccc} \mathbf{A}_{\hat{\mathbf{u}}\hat{\mathbf{u}}} & \mathbf{0} & \mathbf{A}_{\hat{\mathbf{u}}\mathbf{u}} & \mathbf{A}_{\hat{\mathbf{u}}\mathbf{L}} & \mathbf{A}_{\hat{\mathbf{u}}p} & \mathbf{0} \\ \mathbf{A}_{\rho\hat{\mathbf{u}}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{A}_{u\hat{\mathbf{u}}} + \mathbf{C}_{u\hat{\mathbf{u}}}(\hat{\mathbf{u}}) & \mathbf{0} & \mathbf{A}_{uu} + \mathbf{C}_{uu}(\mathbf{u}) & \mathbf{A}_{uL} & \mathbf{A}_{up} & \mathbf{0} \\ \mathbf{A}_{L\hat{\mathbf{u}}} & \mathbf{0} & \mathbf{A}_{Lu} & \mathbf{A}_{LL} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{p\hat{\mathbf{u}}} & \mathbf{0} & \mathbf{A}_{pu} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{\rho p}^T \\ \mathbf{0} & -1 & \mathbf{0} & \mathbf{0} & \mathbf{A}_{\rho p} & \mathbf{0} \end{array} \right] \left\{ \begin{array}{c} \hat{\mathbf{u}} \\ \rho \\ \mathbf{u} \\ \mathbf{L} \\ \mathbf{p} \\ \lambda \end{array} \right\} - \left\{ \begin{array}{c} \bar{\mathbf{t}} \\ \mathbf{0} \\ \bar{\mathbf{s}} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{array} \right\} = \mathbf{0}. \quad (2.14)$$

The nodal values of $\hat{\mathbf{u}}$, \mathbf{u} , \mathbf{L} and p are represented by $\hat{\mathbf{u}}$, \mathbf{u} , \mathbf{L} and \mathbf{p} , respectively. The constraints (2.9d) are applied using the Lagrangian multiplier λ . The dashed lines inside the matrix separates the global and local problems. The elemental matrices used to compute the system are presented in Appendix C.

The equations below the dashed line in (2.14) correspond to the discretization of the local problems (2.9). In the linear case, for the Stokes equations, these can be solved element-by-element to define the so-called local solver, *i.e.*, the expression of the local variables $\mathbf{u}^{(e)}$, $\mathbf{L}^{(e)}$ and $\mathbf{p}^{(e)}$ (where (e) denotes the nodal values for element Ω^e) in terms of global variables $\hat{\mathbf{u}}$ and ρ_e . Replacement of the local solver for each element in the global equations (2.12), *i.e.*, in the equations above the dashed line in (2.14), leads to a system of equations involving only the global variables $\hat{\mathbf{u}}$ and ρ . Figure 2.1b represents the nodes corresponding to the actual unknowns of the global problem of HDG. After the global problem is solved, the solution inside each element, $\mathbf{u}^{(e)}$, $\mathbf{L}^{(e)}$ and $\mathbf{p}^{(e)}$, can be computed with the local solver.

Remark 1. Equation (2.9d) must be scaled by the perimeter of the element, $|\partial\Omega^e|$, to get a symmetric system, in the case of Stokes problem, after condensation of the local variables. Another possibility is choosing the average of the pressure inside each element, which must be scaled by the area of the element.

In the non-linear case, for the Navier-Stokes equations, the solution of the non-linear system with Newton-Raphson leads to a linear system of equations to be solved in each iteration. This linear system can be solved analogously to the Stokes solution, *i.e.*, the local equations are solved element-by-element, to express the local variables in terms of global variables, leading to a smaller linear system of equations involving only trace variables and the mean of the pressure in the elements.

The HDG formulation provides a numerical solution with optimal convergence of order $k + 1$ in \mathcal{L}_2 norm for the velocity, \mathbf{u} , the pressure, p , and also for the approximation of the gradient, \mathbf{L} . In addition, the mean of the velocity in each element, $(\mathbf{u}, 1)_{\Omega^e}$, is super-convergent with errors of order $k + 2$. Hence, a new super-convergent approximation of velocity, \mathbf{u}^* , can be computed by solving a new problem in each element. The local post-processing proposed by Nguyen *et al.* (2010) is used in the present work which requires solving,

$$\begin{aligned} -\operatorname{div}(\operatorname{grad} \mathbf{u}^*) &= \operatorname{div} \mathbf{L} && \text{in } \Omega^e, \\ -(\operatorname{grad} \mathbf{u}^*) \mathbf{n} &= \mathbf{L} \mathbf{n} && \text{on } \partial\Omega^e, \\ (\mathbf{u}^*, 1)_{\Omega^e} &= (\mathbf{u}, 1)_{\Omega^e}. \end{aligned} \quad (2.15)$$

The weak form can be stated in a richer finite dimensional space: find $\mathbf{u}^* \in [\mathcal{V}_{k+1}^h(\Omega)]^2$ such that

$$\begin{aligned} (\operatorname{grad} \delta \mathbf{u}^*, \operatorname{grad} \mathbf{u}^*)_{\Omega^e} &= (\operatorname{grad} \delta \mathbf{u}^*, \mathbf{L})_{\Omega^e}, \\ (\mathbf{u}^*, 1)_{\Omega^e} &= (\mathbf{u}, 1)_{\Omega^e}, \end{aligned} \quad (2.16)$$

for all $\delta \mathbf{u}^* \in [\mathcal{V}_{k+1}^h(\Omega)]^2$ and $e = 1, \dots, n_{el}$, where $\mathcal{V}_{k+1}^h(\Omega)$ is a richer space with one degree more than \mathcal{V}_k^h . The constraint in equation (2.16) is applied using Lagrangian multipliers for solving the system in the each element. The super-convergent velocity, \mathbf{u}^* , converges asymptotically with a rate of $k+2$ in the \mathcal{L}_2 norm for a mesh with uniform degree, k (Cockburn *et al.*, 2011). Convergence properties for the non-uniform degree are discussed in detail in Chen and Cockburn (2012, 2014). The computational overhead in computing the super-convergent solution, \mathbf{u}^* , is small as it is done in element-by-element fashion. This solution can be used to define a reliable and inexpensive error estimator for HDG velocity approximation, \mathbf{u} (Giorgiani *et al.*, 2014).

Remark 2. It is important to apply the Dirichlet boundary conditions using \mathcal{L}_2 projection instead of nodal interpolation. It is noticed that nodal interpolation of Dirichlet data results in the loss of optimal convergence for the gradient of velocity, \mathbf{L} , and eventually the super-convergence of post-processed velocity.

2.3 Static condensation

Static condensation is used in both CG and HDG methods in the present work: the DOFs that are not shared by neighbouring elements can be expressed in terms of remaining DOFs of the element, hence reducing the global DOFs of the system. In the case of CG, interior nodes are not shared by other elements and, therefore, they can be expressed in terms of boundary

nodes of the element. In the case of HDG, all the local DOFs are approximated independently inside each element, consequently, they can be expressed in terms of global DOFs.

Let $\delta \mathbf{x}_m$ and $\delta \mathbf{x}_s$ be the incremental master and slave DOFs in a linearised system. In the case of CG, master and slave DOFs correspond to boundary and interior DOFs respectively, while in HDG, they represent local and global DOFs, respectively. A typical linearised system of equations can be written in the form,

$$\begin{bmatrix} \mathbf{A}_{mm} & \mathbf{A}_{ms} \\ \mathbf{A}_{sm} & \mathbf{A}_{ss} \end{bmatrix} \begin{Bmatrix} \delta \mathbf{x}_m \\ \delta \mathbf{x}_s \end{Bmatrix} = - \begin{Bmatrix} \mathbf{r}_m \\ \mathbf{r}_s \end{Bmatrix}. \quad (2.17)$$

The matrix \mathbf{A}_{ss} is block diagonal and its inverse is well defined. Hence, $\delta \mathbf{x}_s$ can be expressed in terms of $\delta \mathbf{x}_m$ in an element-by-element fashion as,

$$\delta \mathbf{x}_s^{(e)} = \left[-\mathbf{A}_{ss}^{(e)} \right]^{-1} \mathbf{A}_{sm}^{(e)} \delta \mathbf{x}_m - \left[\mathbf{A}_{ss}^{(e)} \right]^{-1} \mathbf{r}_s^{(e)}. \quad (2.18)$$

Note that (e) represents the elemental matrices. Replacing $\delta \mathbf{x}_s$ from equation (2.18) into the first of the (2.17) equations and assembling into the global system results in the following,

$$(\mathbf{A}_{mm} - \mathbf{A}_{ms} \mathbf{A}_{ss}^{-1} \mathbf{A}_{sm}) \delta \mathbf{x}_m = -\mathbf{r}_m + \mathbf{A}_{ms} \mathbf{A}_{ss}^{-1} \mathbf{r}_s. \quad (2.19)$$

As the matrices are condensed on elemental basis, the computational overhead is negligible. In the case of a problem with very high DOF count, this numerical technique can save significant CPU time in solving the system of equations, especially for high degree approximations.

2.4 Implementation details

In all the results presented in the current work the shape functions that are used to approximate the variables inside each element are generated using Fekete nodal distributions (Taylor *et al.*, 2000) for triangular elements and Gauss–Lobatto points (Abramowitz, 1974, p. 888) in the case of quadrilateral elements. The shape functions are computed using Jacobi polynomials (Hesthaven and Warburton, 2002, 2007). All the meshes are generated using EZ4U (Sarrate and Huerta, 2000, 2002, LaCaN, 2016), which is a high order mesh generator, and Gmsh (Geuzaine and Remacle, 2009) is used to post process the results.

The non-linear system of equations is solved using full Newton–Raphson method. Relative incremental and residual norms are used as convergence criteria with a tolerance of 10^{-12} .

An in-house code is implemented in FORTRAN. Only direct solvers are used in the present work. Harwell Subroutine Library (HSL) (HSL, 2016) routines MA57d and MA41d (Arioli *et al.*, 1989) are used for solving symmetric and unsymmetric systems, respectively. Both solvers use Approximate Minimum Degree (AMD) (Amestoy *et al.*, 1996) reordering algorithm to reorder the linear system of equations. MC75d (William, 1984) is used to estimate the condition number of the tangent stiffness matrices, $\kappa(\mathbf{A})$.

All tests were performed on machine equipped with 24 Intel(R) Xeon(R) E5-2620 v2 2.10-2.60 GHz processors and 64 GB of RAM running OpenSUSE 13.1 (x86_64) using a serial implementation. The code was compiled using gfortran 4.8.1.

Table 2.1: Vertices and edges/faces count in terms of number of elements, n_{el} .

Mesh Type	Vertices (n_0)	Edges (n_1)/Faces (n_2)
Triangular	$\frac{n_{el}}{2}$	$\frac{3 n_{el}}{2}$
Quadrilateral	n_{el}	$2n_{el}$
Tetrahedral	$\frac{n_{el}}{6}$	$\frac{7 n_{el}}{6}$
Hexahedral	n_{el}	$3n_{el}$

Table 2.2: Analytical expressions for **ndof** for HDG and CG.

Method	2D	
	Triangular	Quadrilateral
HDG(T_k/Q_k)	$(3k + 4) n_{el}$	$(4k + 5) n_{el}$
CG($T_k T_{k-1}/Q_k Q_{k-1}$)	$\left(\frac{9k - 9}{2}\right) n_{el}$	$(6k - 5) n_{el}$
	3D	
	Tetrahedral	Hexahedral
HDG(T_k/Q_k)	$(3k^2 + 9k + 7) n_{el}$	$(9k^2 + 18k + 10) n_{el}$
CG($T_k T_{k-1}/Q_k Q_{k-1}$)	$\left(4k^2 - \frac{28k}{3} + \frac{41}{6}\right) n_{el}$	$(12k^2 - 18k + 10) n_{el}$

2.5 Count of DOFs

A theoretical count of the approximate number of DOFs (**ndof**), for both HDG (T_k or Q_k) and CG ($T_k T_{k-1}$ or $Q_k Q_{k-1}$), is presented. The hypothesis for developing the expressions are provided in the appendix of Huerta *et al.* (2013). The main assumptions of the analysis are structured uniform mesh and very large number of elements, *i.e.*, number of boundary faces are negligible compared to the interior ones. Using the stated hypothesis, the approximate number of vertices (n_0) and edges (n_1)/faces (n_2) can be expressed in terms of the number of elements that are provided in table 2.1. The detailed explanation on deducing **ndof** count for HDG and CG is presented in Appendix B. The final expressions for **ndof** are provided in table 2.2.

Figure 2.2 shows the ratio of **ndof** of HDG to CG against the degree of approximation, k , in 2D and 3D spaces. In both cases only, **ndof** after static condensation are taken into account.

It is evident from the fig. 2.2a that HDG has less **ndof** than CG for both triangular and quadrilateral elements when $k > 5$ in the case of 2D. This is due to the fact that, even though HDG has more **ndof** for velocity, the pressure **ndof** are condensed to a single scalar

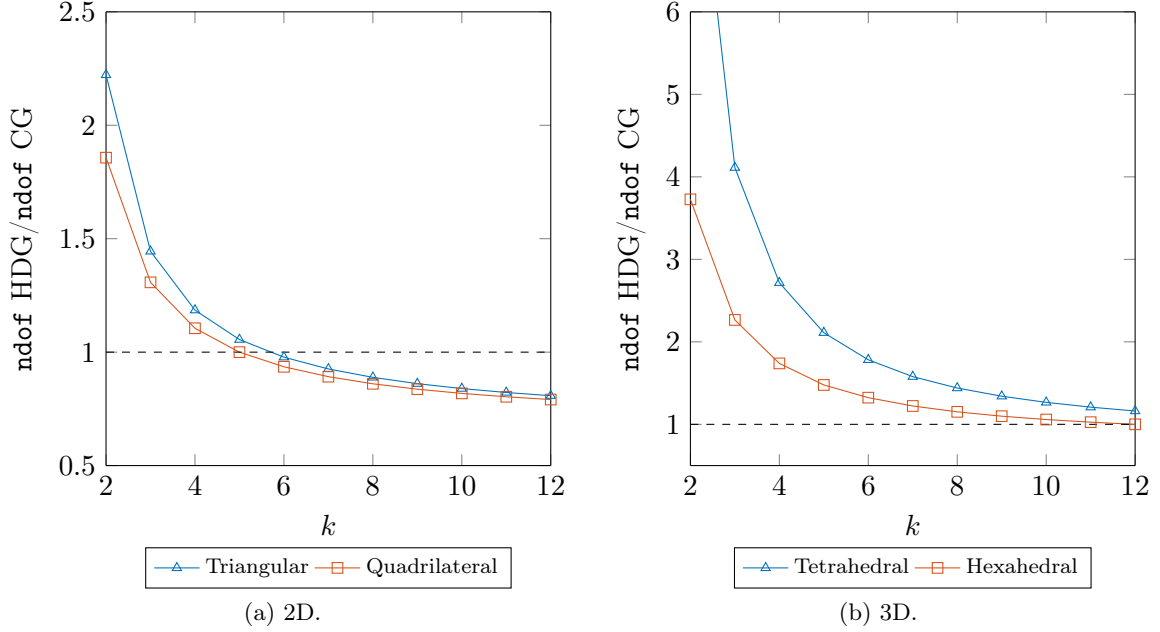


Figure 2.2: Comparison of **ndof** between HDG (T_k/Q_k) and CG ($T_k T_{k-1}/Q_k Q_{k-1}$).

unknown per element, while in CG only interior pressure **ndof** can be condensed out. Hence, at high-degree approximations, HDG tends to have lesser **ndof** than CG. On the other hand, for 3D, HDG has less **ndof** than CG only for very high degrees: $k = 12$ for hexahedral and $k = 19$ for tetrahedral meshes of elements. In this case, both types of elements have more nodes along the edges than vertices in the corresponding 2D case, hence more velocity **ndof** are repeated in HDG. However, the post-processed solution of HDG (T_{k-1}/Q_{k-1}) with degree $k - 1$ has the same order of convergence as CG ($T_k T_{k-1}/Q_k Q_{k-1}$) solution of degree k for velocity. Accordingly, a plot is presented in fig. 2.3, comparing the ratio of number of **ndof** of HDG (T_{k-1}/Q_{k-1}) to CG ($T_k T_{k-1}/Q_k Q_{k-1}$). The ratio favours HDG when $k > 4$ in the case of 2D and, in the case of 3D, $k > 6$ and $k > 12$ for hexahedral and tetrahedral elements, respectively.

Table 2.3 shows the various quantities of interest regarding the linear system of equations for Stokes problem. The number of DOFs of the system is denoted by **ndof** and number of non-zeros of the global stiffness matrix and its factor by **nnz(A)** and **nnz(L)**, respectively. Since, Stokes problems leads to a symmetric matrix, only lower triangular part of the matrix is stored. All the numbers are provided only for free DOFs excluding DOFs corresponding to Dirichlet boundary. From the table, it can be verified that the number of DOFs of HDG is less than CG for $k > 5$, as deduced from theoretical count in fig. 2.2a. Nevertheless, the number of non-zeros in the global matrix and its factor are very similar for $k = 4$ and perhaps, HDG system leads to fewer entries from $k \geq 5$. At higher degrees the entries in the factors of CG systems are almost 1.5 times more than HDG systems. This can favour the HDG systems when using the direct solvers.

It is also worth noting that HDG matrices have a regular block sparsity pattern that is beneficial for the linear solver (Kirby *et al.*, 2012) for the Laplace equation. Consider a mesh with triangular elements: each face has contributions from 4 other faces, as shown in fig. 2.1b. Hence, each row in the final system of HDG has 5 blocks of equal size for velocity DOFs. In

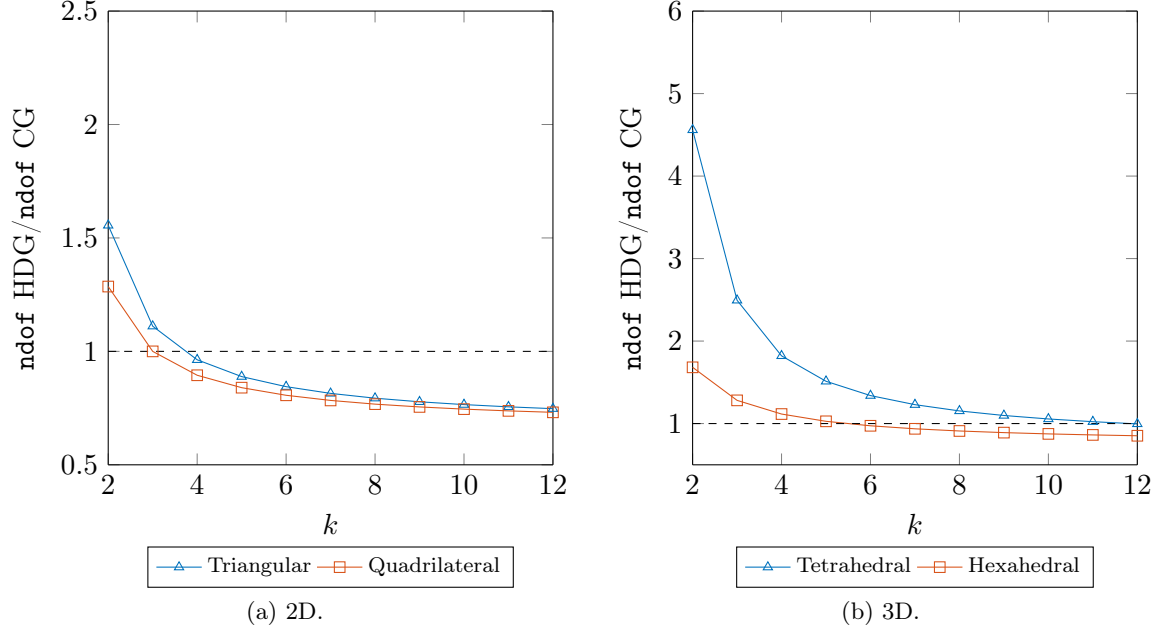
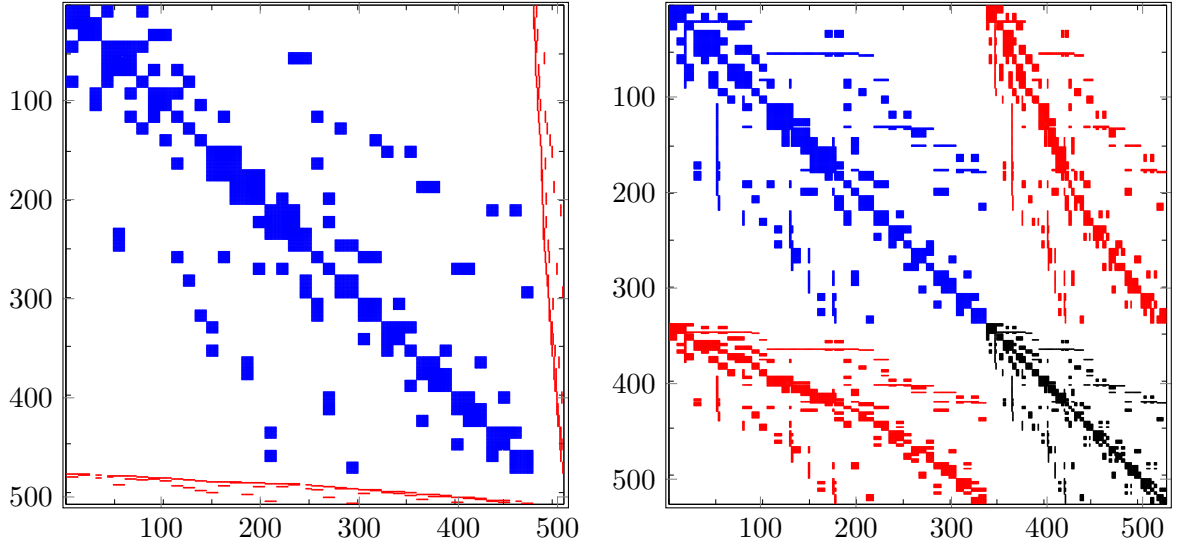

 Figure 2.3: Comparison of ndof between HDG (T_{k-1}/Q_{k-1}) and CG ($T_k T_{k-1}/Q_k Q_{k-1}$).

 Table 2.3: Comparison of DOFs (ndof), number of non-zeros (nnz) in global stiffness matrix (**A**) and its factor (**L**) for Stokes problem with $n_{el} = 2048$ triangular elements.

	k	CG($T_k T_{k-1}$)	HDG(T_k)	k	CG($T_k T_{k-1}$)	HDG(T_k)
ndof	2	9 029	20 095	6	45 634	44 159
nnz(A)		69 641	313 267		2 007 486	1 556 131
nnz(L)		133 866	1 484 136		9 389 209	7 225 163
ndof	3	18 178	26 111	7	54 786	50 175
nnz(A)		400 191	535 279		2 812 691	2 014 687
nnz(L)		1 955 880	2 437 997		12 741 010	9 000 060
ndof	4	27 320	32 127	8	65 938	56 191
nnz(A)		801 236	816 427		3 752 616	2 532 379
nnz(L)		3 755 471	3 751 615		17 006 063	11 344 166
ndof	5	36 482	38 143	9	73 090	62 207
nnz(A)		1 337 001	1 156 711		7 827 261	3 109 207
nnz(L)		6 276 872	5 347 337		21 918 136	13 959 296



(a) Sparsity pattern of HDG, $\text{ndof} = 511$, $\text{nnz} = 1830$. (b) Sparsity pattern of CG, $\text{ndof} = 530$, $\text{nnz} = 3580$.

Figure 2.4: Sparsity pattern of HDG (T_k) and CG ($T_k T_{k-1}$) matrices for $k = 5$ and $h = 1/2$. Variable nnz represents number of non-zeros.

fig. 2.4, the sparsity pattern of matrices of HDG and CG are shown for a regular mesh with 32 triangular elements. The blue fill corresponds to velocity-velocity, the black fill corresponds to pressure-pressure, in the case of CG. The red fill denotes velocity-pressure, in the case of CG, and velocity-mean pressure, in the case of HDG.

2.6 Comparison of computational efficiency

2.6.1 Kovasznay flow

The benchmark problem Kovasznay flow is considered for the comparison of CG and HDG for the solution of Stokes and Navier–Stokes problems. Kovasznay flow is an analytical solution of Navier–Stokes equations in a domain $[0, 2] \times [-0.5, 1.5]$ (Kovasznay, 1948).

$$\begin{aligned} \mathbf{u} &= \begin{bmatrix} 1 - \exp(\lambda x_1) \cos(2\pi x_2) \\ \frac{\lambda}{2\pi} \exp(\lambda x_1) \sin(2\pi x_2) \end{bmatrix}, \\ p &= -\frac{1}{2} \exp(2\lambda x_1) + C, \end{aligned} \quad (2.20)$$

where $\lambda = \frac{Re}{2} - \sqrt{\frac{Re^2}{4} + 4\pi^2}$ and $Re = \frac{1}{\nu}$ is the Reynolds number. The analytical velocity and pressure are shown in fig. 2.5. Dirichlet boundary conditions are prescribed for the velocity on all the boundary. In the case of Stokes problem, a body force equal to the convective term, $\text{div}(\mathbf{u} \otimes \mathbf{u})$, is set using the exact solution at $Re = 20$.

Mesheres are obtained by splitting a regular $n \times n$ Cartesian grid into either $2n^2$ triangles or n^2 quadrilaterals, which gives an uniform element size, $h = 2/n$. All the results are presented considering the stability parameter $\tau_u = 1$ on all faces of each element.

In this section, results are presented for the Stokes problem, and just commented for the Navier–Stokes case.

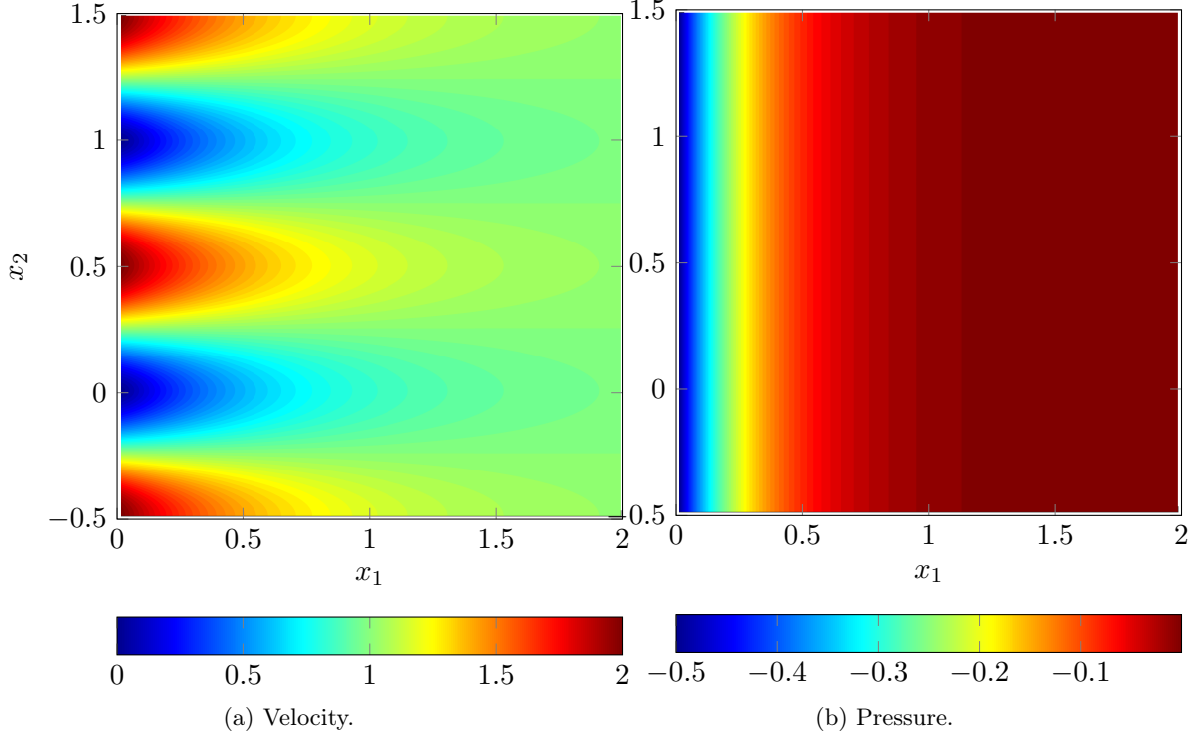


Figure 2.5: Kovaszny flow: analytical solution.

Convergence

The \mathcal{L}_2 norm of the error in the post-processed velocity of HDG is compared to the \mathcal{L}_2 norm of error in velocity of CG. Similarly, the \mathcal{L}_2 norm of the error in pressure is also compared between HDG and CG.

Figure 2.6 shows the convergence plots, for triangular elements, of velocity for HDG and CG, while fig. 2.7 has the convergence results of pressure. In HDG, both velocity and pressure are approximated with the same degree of approximation, k , while in the case of CG, degree k is used for velocity and $k - 1$ for pressure. The HDG post-processed velocity converges with order $k + 2$, compared to $k + 1$ for CG. Similarly, HDG has $k + 1$ rate for pressure compared to the rate of k for pressure of CG.

Although not presented, optimal theoretical convergence with similar accuracy is observed for Navier–Stokes problem in both HDG and CG.

CPU time for linear solver

CPU times for linear solver (in seconds) are presented, for both HDG and CG, to compare their computational efficiency. The time taken for pre-processing, computation and assembly of matrices and post-processing is highly implementation-dependent and hence, not taken into account. The errors considered for comparison are the \mathcal{L}_2 norm of error in post-processed velocity in the case of HDG, with degree k for all variables, and the \mathcal{L}_2 norm of error in velocity of CG, with degree k for velocity and $k - 1$ for pressure.

The results are divided into two groups, namely, low-degree, from 2 to 5, and high-degree, from 6 to 9. The element sizes, h , used for the low-degree results, are $2/\{2^3, 2^4, 2^5, 2^6\}$, while

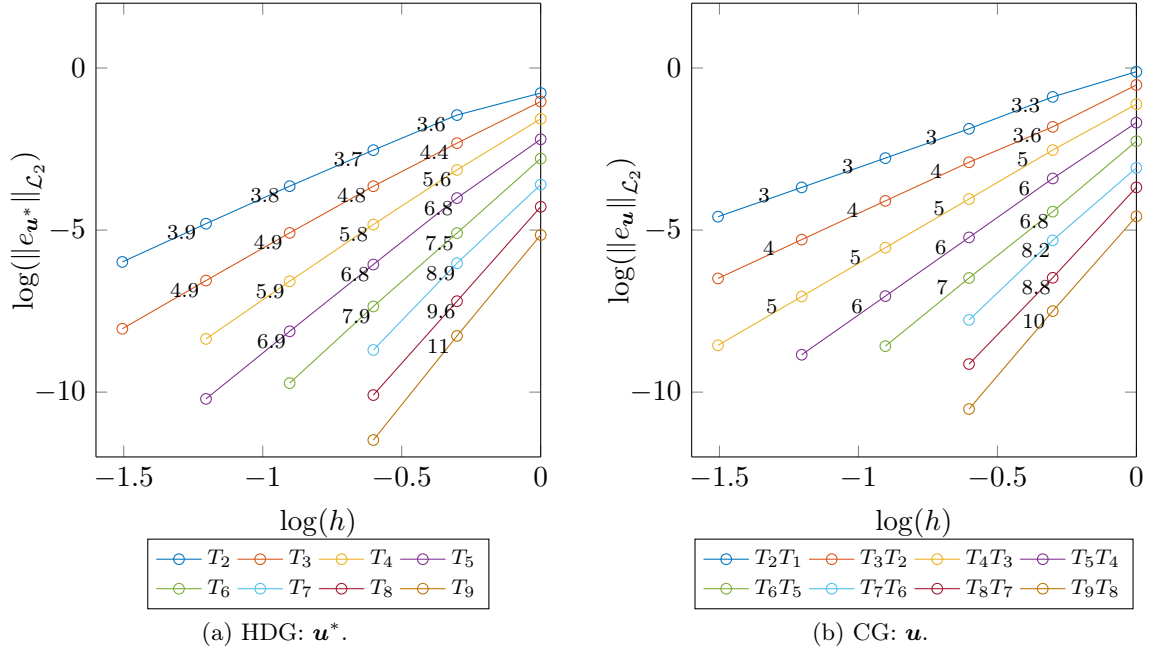


Figure 2.6: Kovaszny flow: convergence of velocity.

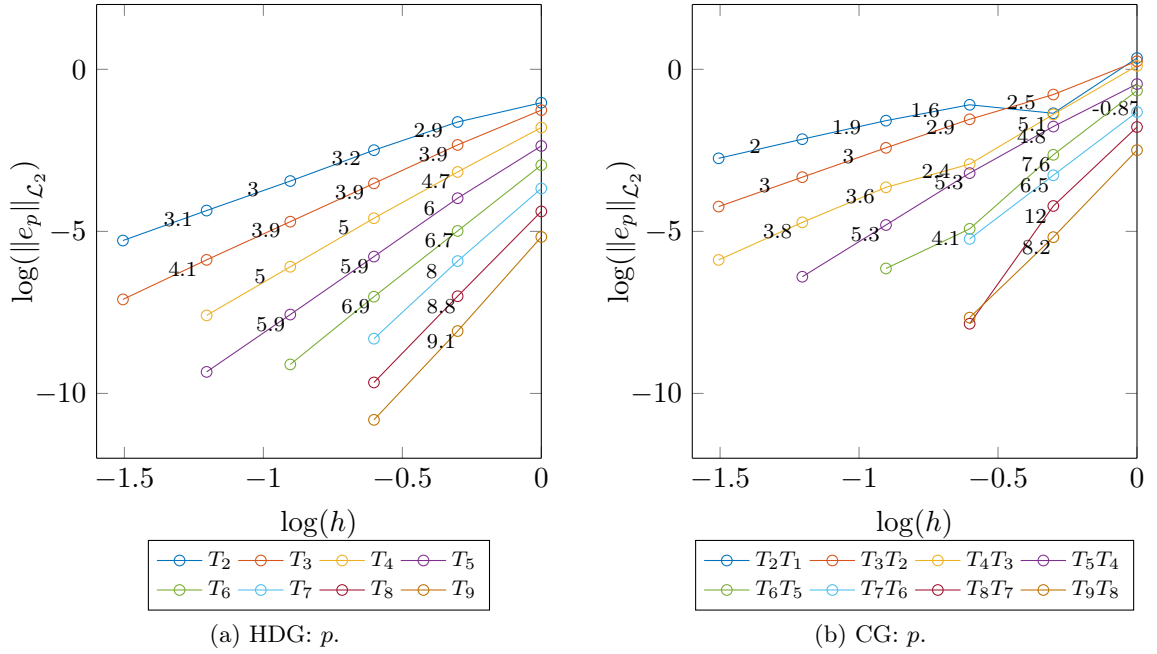


Figure 2.7: Kovaszny flow: convergence of pressure.

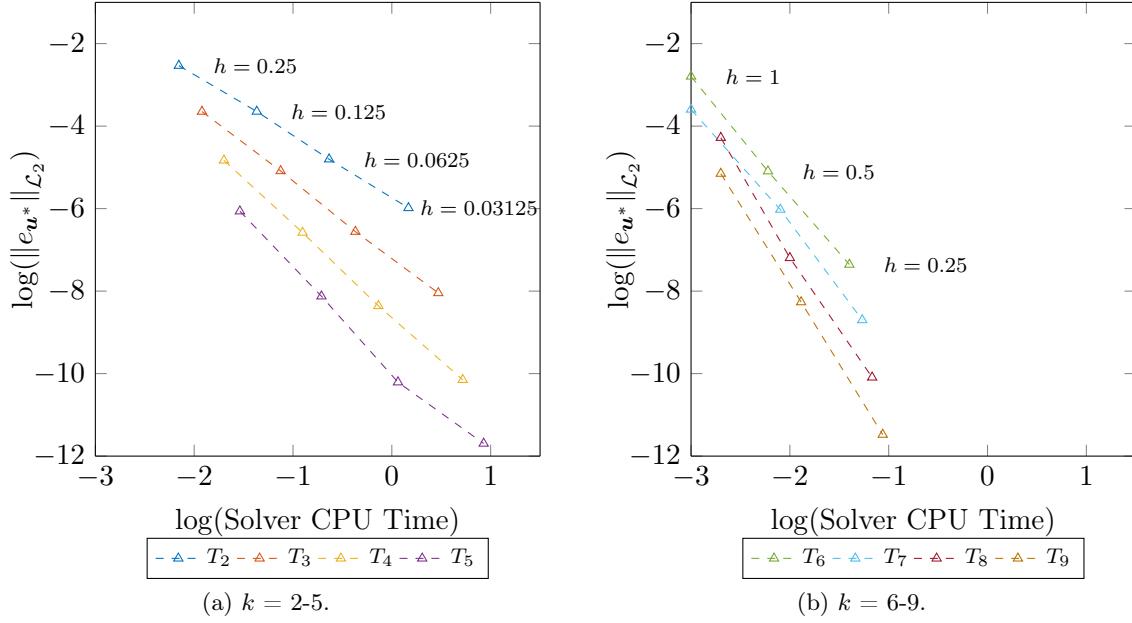


Figure 2.8: Kovasznay flow: error for HDG *vs.* CPU time for linear solver with triangular elements.

coarser meshes, with element sizes $2/\{2^1, 2^2, 2^3\}$, are used in high-degree computations, to keep error within acceptable bounds.

Figures 2.8 and 2.9 show the CPU time taken by the linear solver for various degrees of approximation, for both HDG and CG, in the case of Stokes problem with triangular elements. A common trend can be observed from the plots: to achieve the same level of accuracy, it is more computationally efficient to use a high-degree coarser mesh than a low-degree finer mesh. This may be due to the fact that the data dependencies in a high-degree mesh between elements are lesser than in a low-degree mesh (Yakovlev *et al.*, 2016). A similar trend is observed in the case of Navier–Stokes results and for quadrilateral elements, hence, the plots are omitted.

Figure 2.10 presents a similar plot for a given mesh, with element size $h = 1/2^5$ for $k = 2 - 5$, and with $h = 1/2^2$ for $k = 6 - 9$, comparing the efficiency of HDG and CG in a single plot. Asterisk (*) on HDG denotes the post-processed solution for velocity. For a given level of error, HDG always outperforms CG at all the degrees of approximation presented, and the performance gap between CG and HDG increases with increasing degree of approximation. A similar trend in results is obtained for Navier–Stokes problem, but with a greater difference in the CPU times between HDG and CG at $k = 5$.

Figures 2.11 and 2.12 show the ratio of CPU times of HDG to CG, for triangular elements and quadrilateral elements, respectively. As previously, low-degree plots consider h between 0.25 and 0.03125, while high-degree plots have h between 0.5 and 0.25, to keep errors larger than machine precision. The results of high-degree meshes with $h = 1$ are omitted because their CPU times are too small to be reliable. It can be observed from these plots that, for degree $k \geq 4$, most of the points lie below the ratio of 1, indicating that HDG takes lesser CPU time for linear solver. In the case of high-degree, the CPU times are in the order of 1 second and hence, the ratios are more fluctuating compared to low-degree plots, where CPU times are bigger and ratios are more reproducible and stable. Nevertheless, the ratios stay

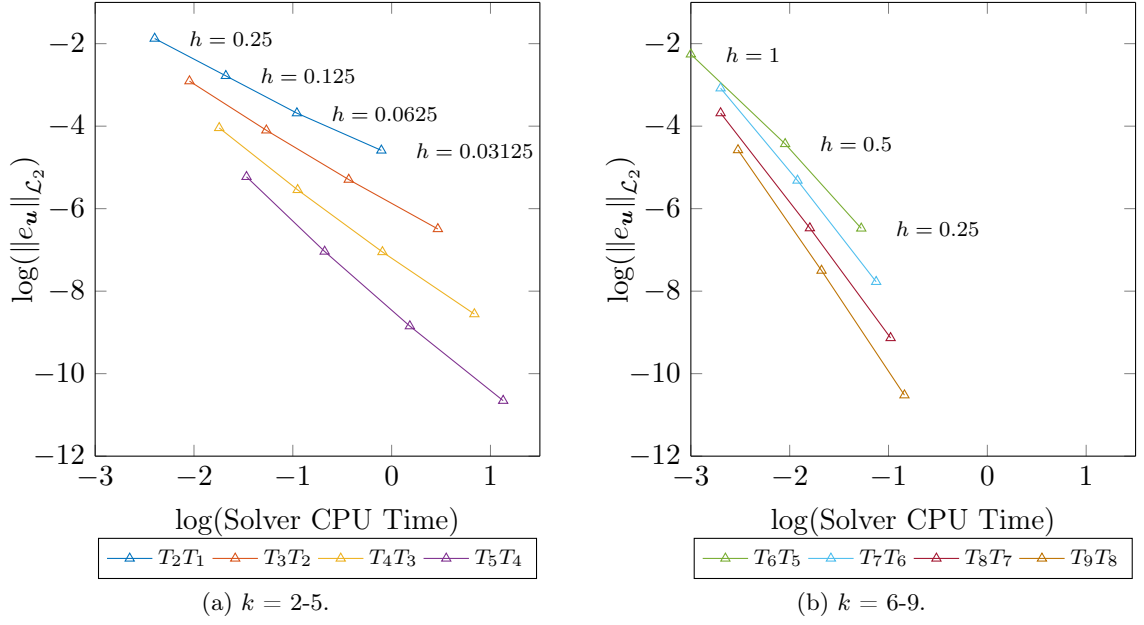


Figure 2.9: Kozaszny flow: error for CG *vs.* CPU time for linear solver with triangular elements.

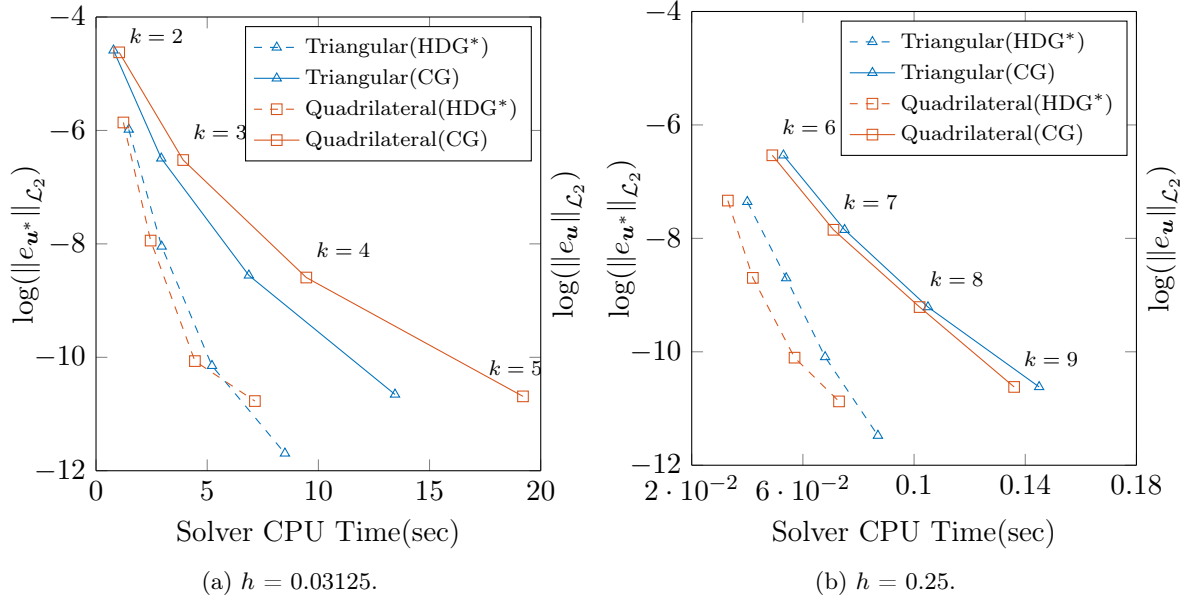


Figure 2.10: Kozaszny flow: comparison of error between HDG and CG *vs.* CPU time for linear solver.

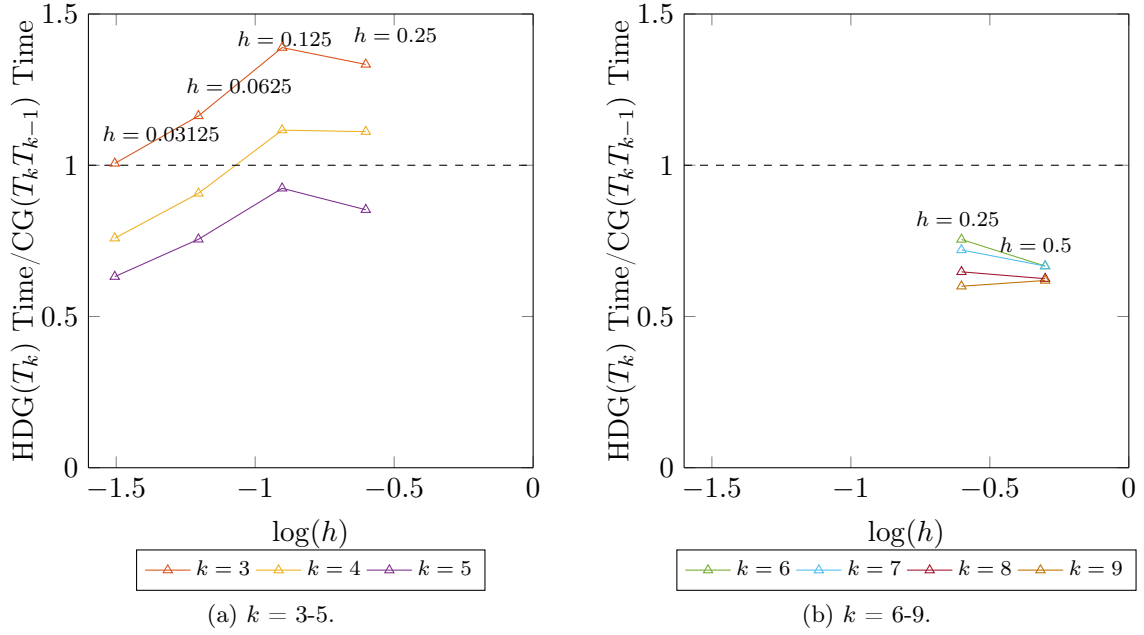


Figure 2.11: Kovasznay flow: ratio of CPU times for linear solver *vs.* element size for triangular elements.

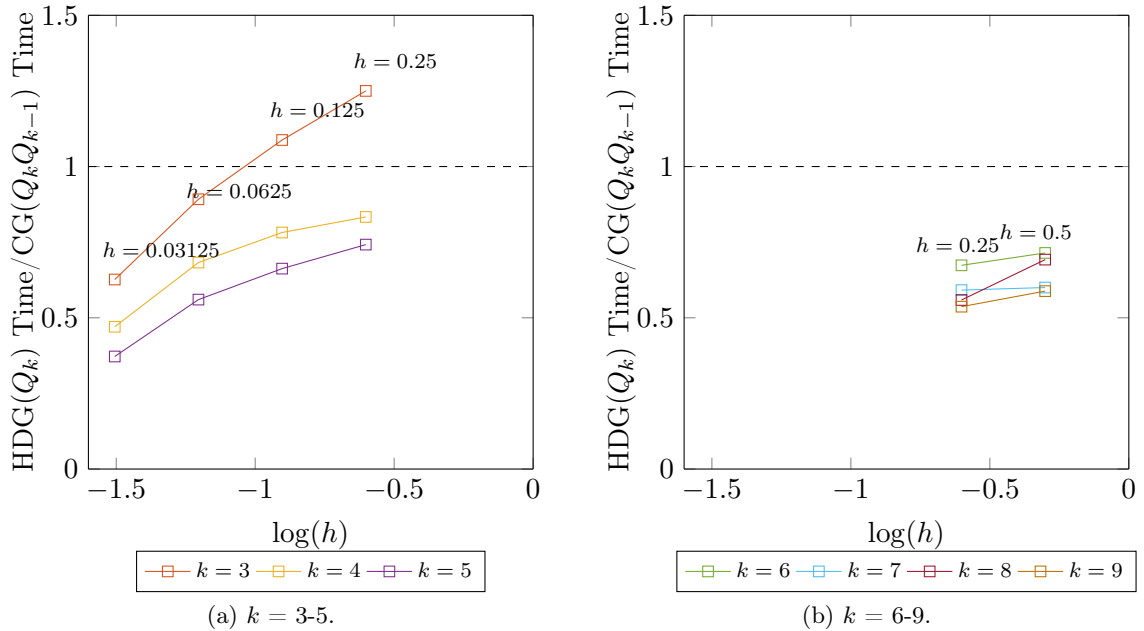


Figure 2.12: Kovasznay flow: ratio of CPU times for linear solver *vs.* element size for quadrilateral elements.

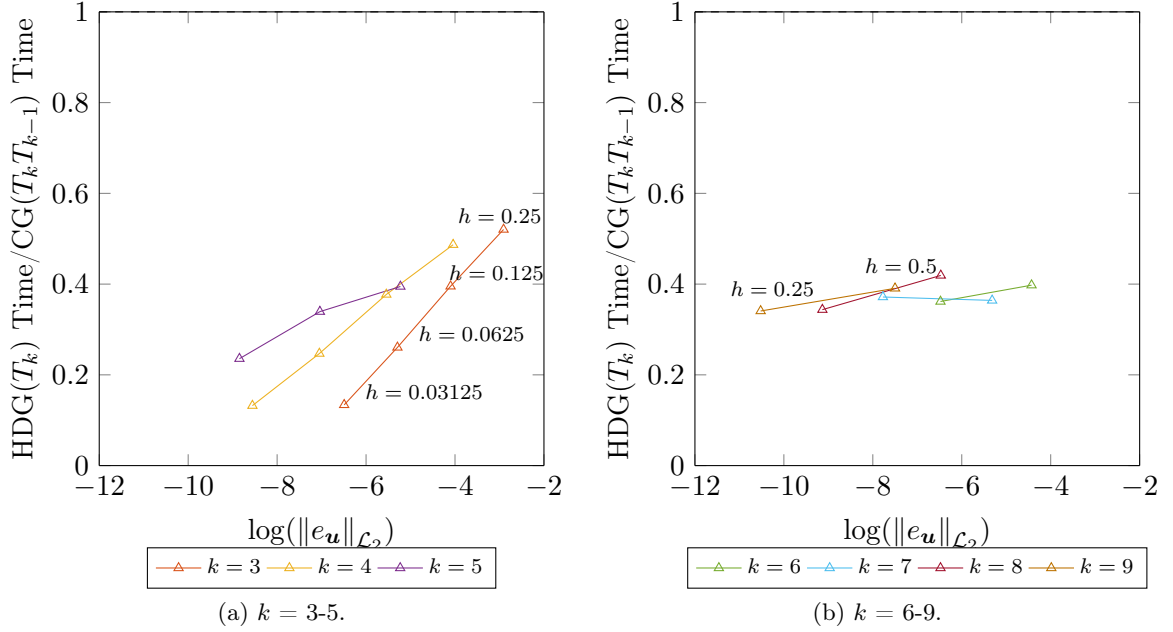


Figure 2.13: Kovasznay flow: ratio of CPU times for linear solver *vs.* error for triangular elements.

below 1 in the case of high-degree suggesting HDG is competitive.

Moreover from the convergence plots in fig. 2.6, it is concluded that the HDG produces lesser error compared to CG for same mesh and same degree of approximation. Hence, a comparison of CPU time against the error is a fairer comparison.

Figures 2.13 and 2.14 show the ratio of CPU times against error. These plots are produced using the data of convergence plots in fig. 2.6 and the CPU times for linear solver in figs. 2.8 and 2.9. For a given degree of approximation and mesh, HDG produces lesser error than CG, thus, for every mesh in the CG plot, the CPU time of HDG is interpolated from figure 2.8 to determine the CPU time that would provide the same error as CG. HDG is more efficient than CG for any given accuracy in both triangle and quadrilateral cases. In the case of high-degree plots, ratios are more fluctuating, again because CPU times are too low.

In the case of Navier–Stokes, a similar trend is observed, with ratios below 1. Hence, the results are not presented to avoid redundancy.

Finally, fig. 2.15 shows the variation of condition number of global stiffness matrix of the Stokes problem, $\kappa(\mathbf{A})$, with degree of approximation, k , for a mesh with triangular elements and $h = 1/2^5$. The condition number increases more rapidly in the case of CG than HDG: condition number in HDG increases by one order of magnitude when going from degree 2 to 9, whereas in CG it increases by approximately 5 orders.

2.6.2 NACA airfoil

The results presented until this point are for regular uniform meshes with a benchmark problem. Now, a more practical problem is considered in this section: the computation of the lift coefficient for the NACA0012 airfoil section at Reynolds number $Re = 5000$ and angle of attack $\alpha_a = 2^\circ$. Steady state Navier–Stokes equations are solved and the error in the lift

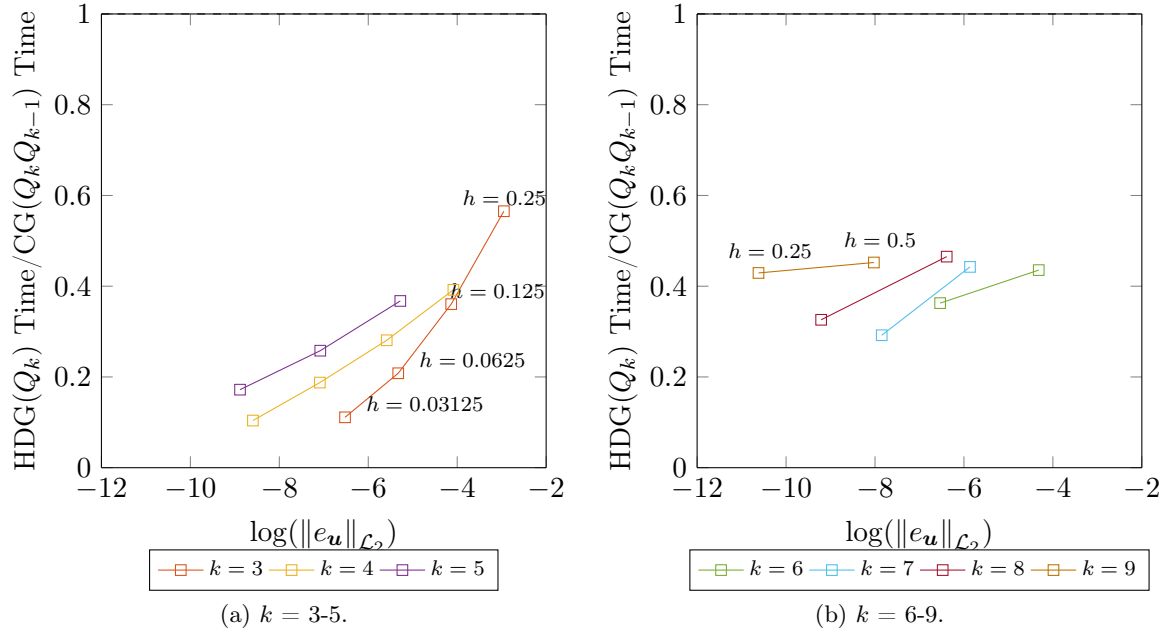


Figure 2.14: Kovasznay flow: ratio of CPU times for linear solver *vs.* error for quadrilateral elements.

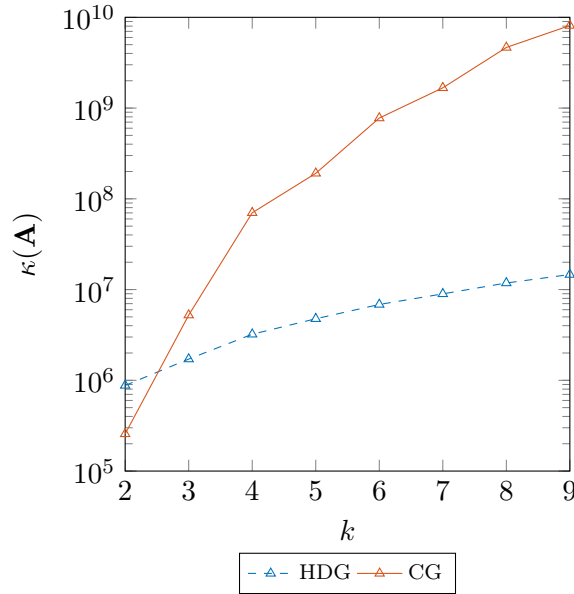


Figure 2.15: Kovasznay flow: Condition numbers in Stokes problem with triangular elements and $h = 1/2^5$.

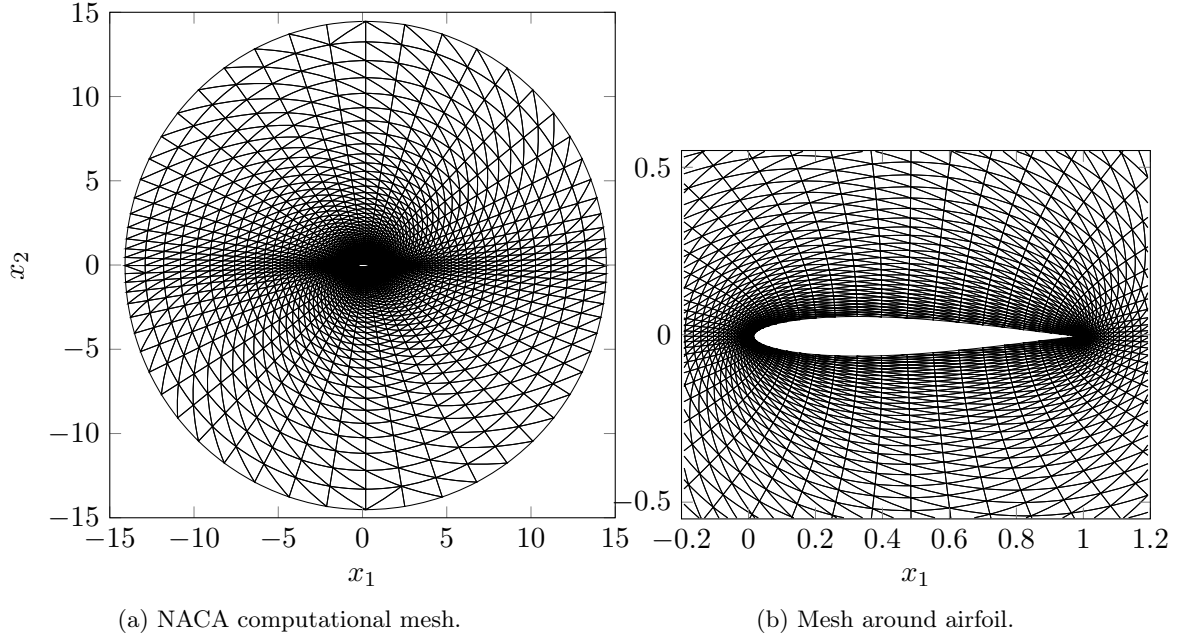


Figure 2.16: NACA airfoil: computational mesh (refinement level 4) and zoom.

coefficient is used for presenting the results.

The meshes are generated using an in-house code for a computational domain with a circular exterior boundary. Four different meshes are used in the computations with nested refinement. Figure 2.16 shows the most refined mesh used in the computations and the region around the airfoil.

The mesh is non-uniform with refinement in the vicinity of the airfoil section. Depending on the angle of attack, α_a , the velocity on the inflow half of the boundary is prescribed to $(\cos \alpha_a \mathbf{e}_1 + \sin \alpha_a \mathbf{e}_2)$. The rest of the exterior boundary is treated as outflow boundary, as it is far from the airfoil. No slip boundary condition is applied along the boundary of the airfoil.

The velocity field around the airfoil, obtained using the mesh presented in fig. 2.16 and degree of approximation $k = 5$, is shown in fig. 2.17. The singularity at the front tip of the airfoil can be noticed, and it can be observed that the wake region of the airfoil is steady at this Reynolds number without any vortices.

The results of computational efficiency with respect to CPU times are presented next considering the error in lift coefficient. More specifically, CPU times reported are the average CPU time for each Newton–Raphson iteration and the reference value for the lift coefficient is obtained with a CG computation with the mesh shown in fig. 2.16 and degree $k = 8$. In both HDG and CG, the non-linear solver takes around 5-6 Newton–Raphson iterations for attaining convergence with tolerance 10^{-12} .

Figure 2.18 shows the ratio of the CPU times for the linear solver *vs.* element size. The element size, h , in the case of NACA test is taken as the size of the biggest element in the mesh. All the ratios lie below 1, leading to the conclusion that HDG is more efficient than CG. Additionally, in this test, the ratios are nearly constant for each degree of approximation. The computational domain in this example is relatively big and hence, larger CPU times are noticed, which are more reliable and reproducible than for the Stokes problem for Kovasznay flow.

CPU time with error is plotted in fig. 2.19 by interpolating the CPU times of HDG as stated

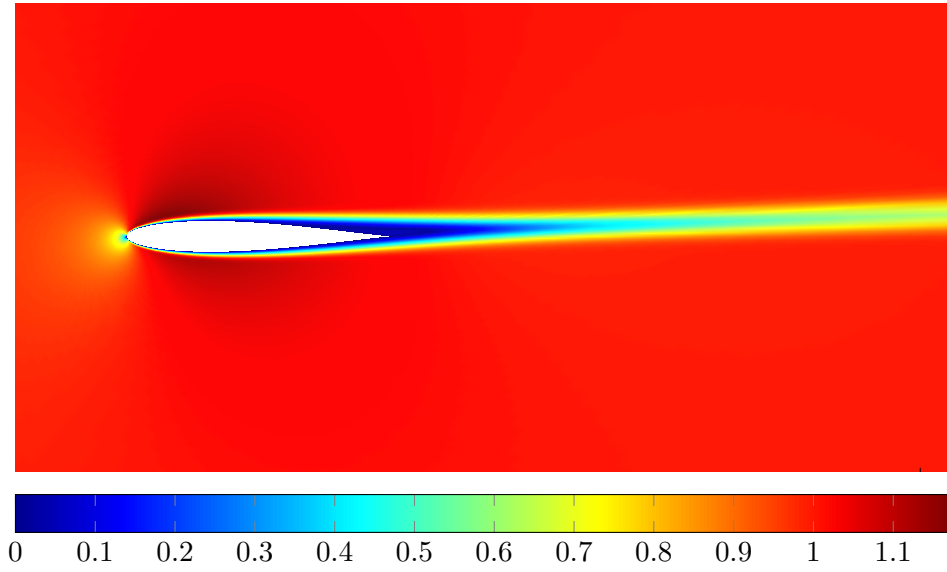


Figure 2.17: NACA airfoil: velocity field with $k = 5$ and refinement level 4.

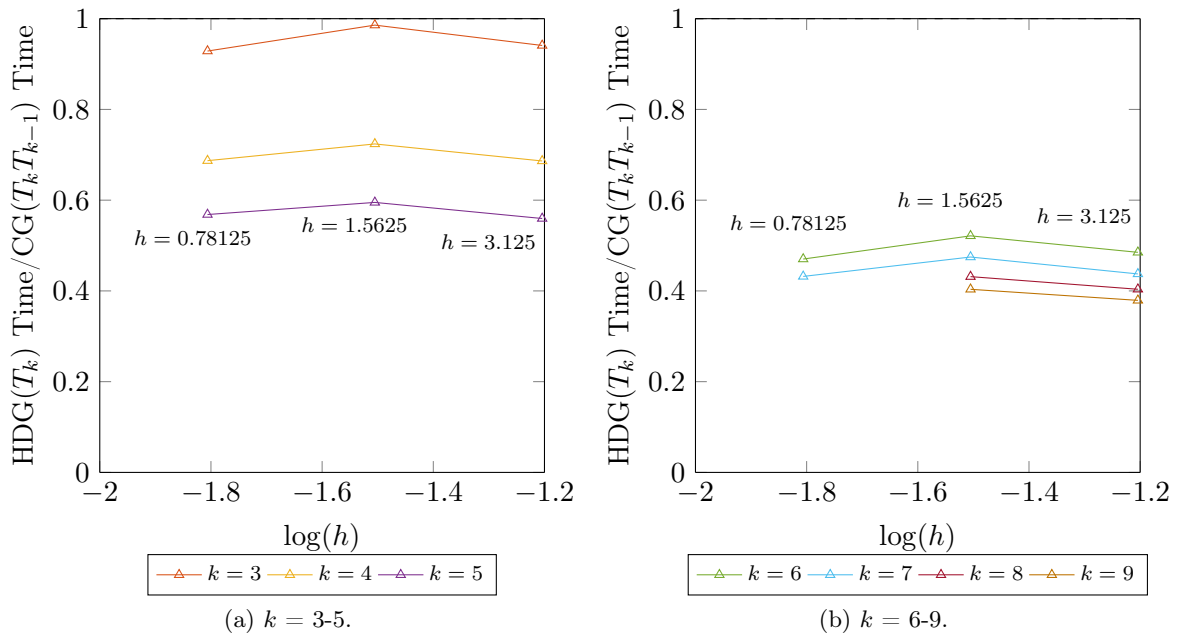


Figure 2.18: NACA airfoil: ratio of CPU times for linear solver *vs.* element size for triangular elements.

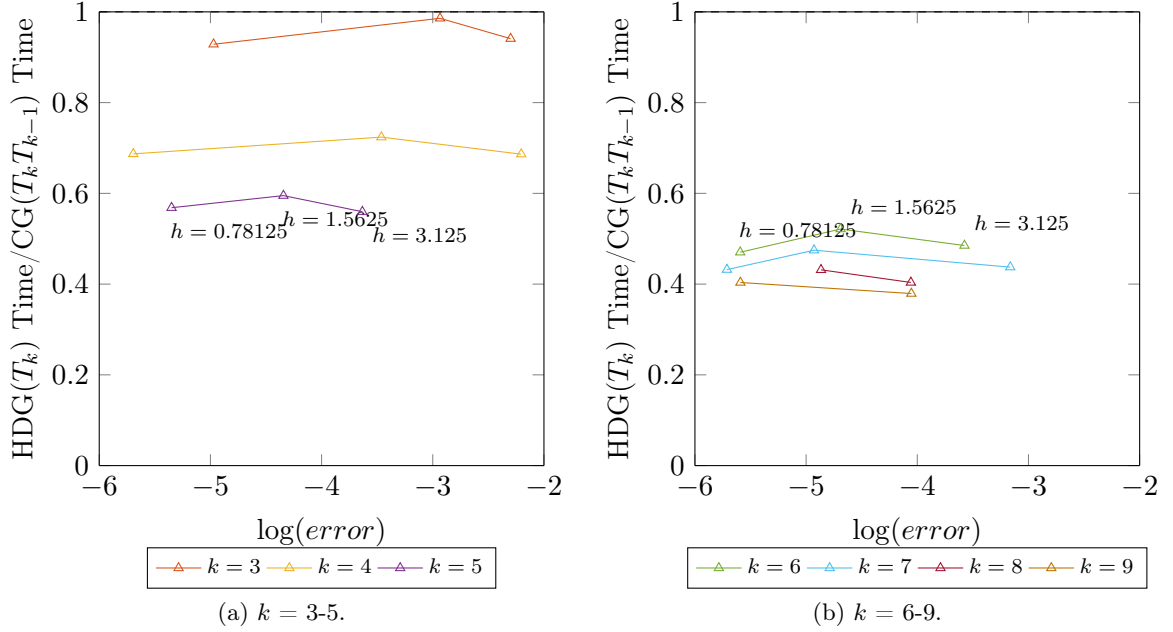


Figure 2.19: NACA airfoil: ratio of CPU times for linear solver *vs.* error for triangular elements.

in Kovasznay flow section. Again, all the ratios stay below 1, leading to similar conclusions: HDG is more computationally efficient than CG when CPU times for the linear solver are considered.

Finally, fig. 2.20 shows the maximum condition number in Newton–Raphson iterations in HDG and CG. All the condition numbers presented are for triangular elements and $h = 0.78125$. On the one hand, in the case of HDG, condition number remains practically constant with increasing degree of approximation. On the other hand, in the case of CG, there is no particular trend observed, except that for the cases $k = \{2, 6\}$, $\kappa(\mathbf{A})$ is several orders of magnitude higher, relatively to HDG.

2.7 Comparison of stability in the presence of sharp fronts

In this section, the robustness of HDG and CG is compared in terms of stability. The problem chosen to make this study was first reported in Berrone (2001). The analytical solution of this manufactured problem is given as follows,

$$\begin{aligned}
 \mathbf{u} &= \begin{bmatrix} \left(1 - \cos\left(\frac{2\pi(\exp(R_1 x_1) - 1)}{\exp(R_1) - 1}\right)\right) \sin\left(\frac{2\pi(\exp(R_2 x_2) - 1)}{\exp(R_2) - 1}\right) \frac{R_2}{2\pi} \frac{\exp(R_2 x_2)}{(\exp(R_2) - 1)} \\ - \left(1 - \cos\left(\frac{2\pi(\exp(R_2 x_2) - 1)}{\exp(R_2) - 1}\right)\right) \sin\left(\frac{2\pi(\exp(R_1 x_1) - 1)}{\exp(R_1) - 1}\right) \frac{R_1}{2\pi} \frac{\exp(R_1 x_1)}{(\exp(R_1) - 1)} \end{bmatrix}, \\
 p &= R_1 R_2 \sin\left(\frac{2\pi(\exp(R_2 x_2) - 1)}{\exp(R_2) - 1}\right) \sin\left(\frac{2\pi(\exp(R_1 x_1) - 1)}{\exp(R_1) - 1}\right) \frac{\exp(R_1 x_1) \exp(R_2 x_2)}{(\exp(R_1) - 1)(\exp(R_2) - 1)},
 \end{aligned} \tag{2.21}$$

where R_1 and R_2 are two positive parameters. The body force is computed from the Navier–Stokes equations using the analytical solution (2.21). The problem is solved in the computa-

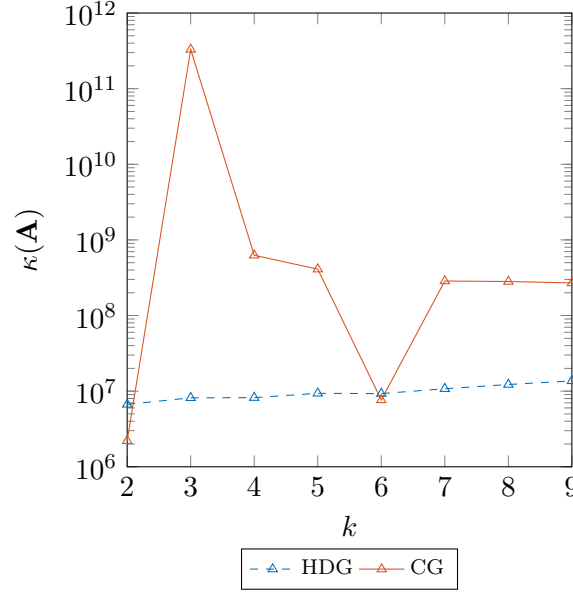


Figure 2.20: NACA airfoil: condition numbers with triangular elements and $h = 0.78125$.

tional domain $[0, 1] \times [0, 1]$.

The velocity field of this solution is similar to a counter clockwise vortex. If (x_0, y_0) are the coordinates of the center of vortex, then the relation between R_1 and R_2 with (x_0, y_0) is

$$x_0 = \frac{\log((\exp(R_1) + 1)/2)}{R_1}, \quad y_0 = \frac{\log((\exp(R_2) + 1)/2)}{R_2}. \quad (2.22)$$

By increasing R_1 , the center goes towards the right side ($x_1 = 1$) of the domain, while by increasing R_2 the center approaches the top edge ($x_2 = 1$) of the domain. In the present study, R_2 is fixed at 0.1 which gives $y_0 = 0.5125$. Hence, the center of vortex is on the line $y_0 = 0.5125$ and, in this case, its distance to the right side is $1/\sqrt[4]{Re}$, where $Re = 1/\nu$. R_1 is chosen in such a way that it satisfies the equation $1/R_1 \log((\exp(R_1) + 1)/2) = 1 - (1/\sqrt[4]{Re})$. Thus, a boundary layer is formed near the right handed edge ($x_1 = 1$) of the domain, which enables to study the relative stability between HDG and CG. Note that this solution is non-symmetric as the line of symmetry is $x_2 = 0.5125$, which does not coincide with the line of symmetry of the domain. Hence, the numerical solutions obtained for the present problem are non-symmetric too.

To make the study, a regular mesh with triangular elements is chosen, with degree of approximation $k = \{3, 5\}$, on a mesh of $h = 1/2^5$, which is relatively fine. Navier–Stokes equations are solved using Dirichlet boundary conditions on all the boundary, computed from the analytical solution, and also including the corresponding body force. In the case of HDG, the stabilization parameter, τ_u , is varied to study its influence on stability. The initial guess for the non-linear solver is set as the nodal values of the analytical solution, to ensure the convergence of non-linear solver in the absence of numerical instabilities.

Figure 2.21 shows the solution for $Re = 2000$. HDG provides a smooth solution using $\tau_u = 1$ with good resolution of boundary layer, whereas no convergence is obtained with CG. Figure 2.21b shows the CG solution of Oseen equations (linearised Navier–Stokes equations) with convection velocity equal to the analytical velocity. It is evident between figs. 2.21a and 2.21b that numerical instabilities exist in CG, precluding convergence of the non-linear solver,

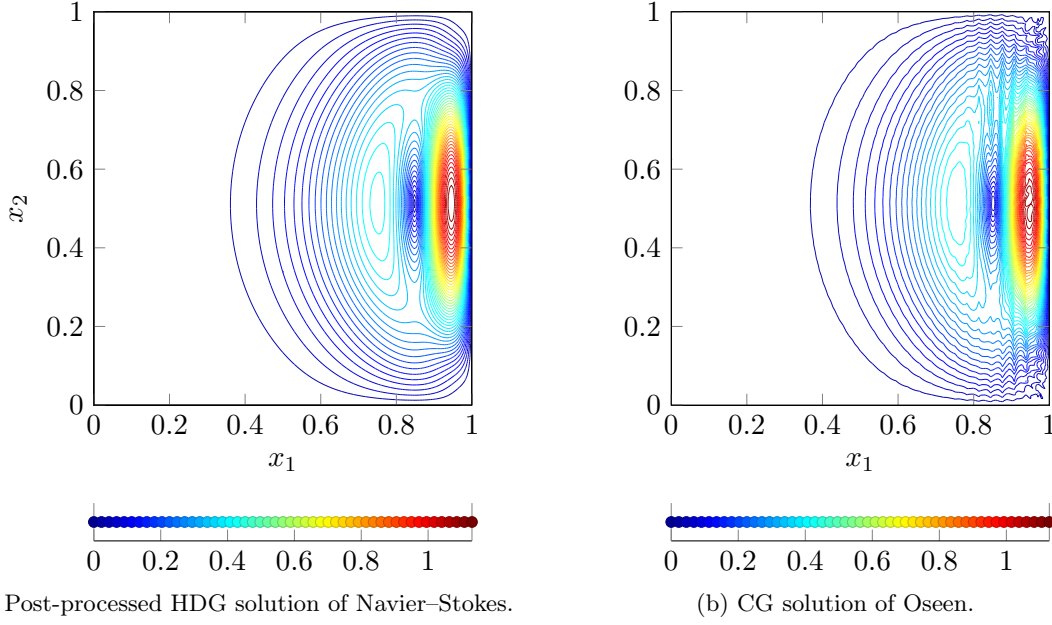


Figure 2.21: Stability study: isolines of the modulus of the velocity field at $Re = 2000$ and $k = 3$.

while HDG does not present any instabilities. Figure 2.22 shows the isolines of the modulus of the velocity field for $Re = 11\,100$ and degree $k = 3$. In the case of HDG, the solution of the Navier-Stokes equations is presented, for $\tau_u = 350$, whereas, the solution the Oseen equations is shown in the case of CG. The CG solution for the Oseen equations draws the similar conclusion as the previous example: the numerical instabilities pollute the solution in the whole domain, precluding convergence of the non-linear solver for the Navier-Stokes equations. In fig. 2.23, the relative residual norm *vs.* the iteration number is presented for both cases, showing that non-linear solver does not converge for CG, whereas in HDG case it exhibits the typical quadratic convergence in the asymptotic limit.

The maximum Reynolds number that can be reached for different degree, k , in the case of HDG and CG, is presented in table 2.4, showing that HDG is more stable than CG for the same degree of approximation and mesh. The stabilization parameter, τ_u , has a considerable influence on the stability of the method. However, for $Re > 11\,100$, increasing τ_u further does not improve the stability properties of the HDG and no convergence is obtained for this mesh and degree $k = 3$.

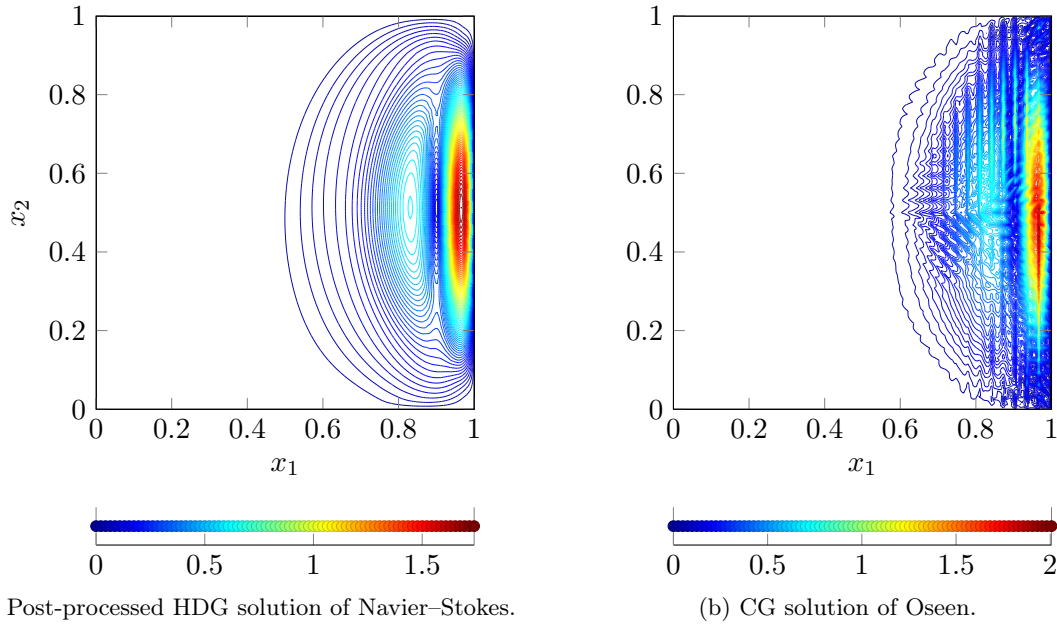


Figure 2.22: Stability study: isolines of the modulus of the velocity field for $Re = 11\,100$ and $k = 3$.

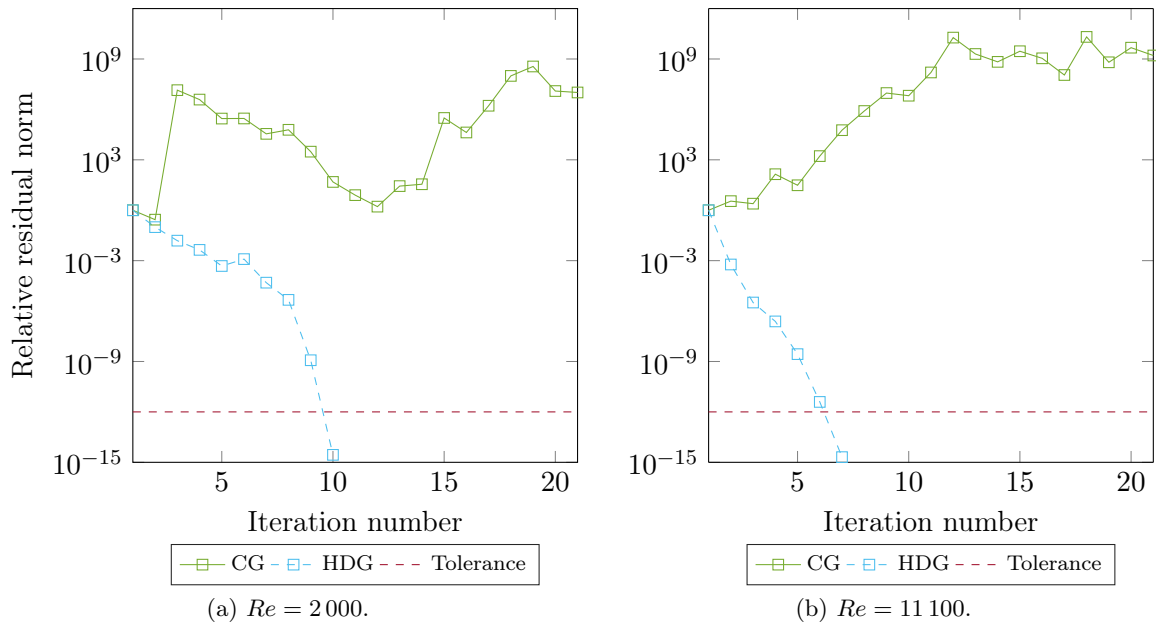


Figure 2.23: Stability study: relative residual norm *vs.* iteration number.

Table 2.4: Maximum Reynolds number for which convergence is achieved, for HDG and CG, for different degree, k .

Method		Degree	
		$k = 3$	$k = 5$
CG		$Re = 1\,870$	$Re = 8\,300$
HDG	τ_u		
	1	$Re = 2\,250$	$Re = 2\,450$
	100	$Re = 8\,000$	$Re = 9\,100$
	350	$Re = 11\,100$	$Re = 15\,400$

2.8 Time discretization using Backward Differentiation Formulae (BDF)

2.8.1 Unsteady incompressible Navier–Stokes equations

The unsteady incompressible Navier–Stokes equations can be written as follows,

$$\begin{aligned}
 \frac{\partial \mathbf{u}}{\partial t} + \operatorname{div}(\mathbf{u} \otimes \mathbf{u}) - \operatorname{div}(-p\mathbf{I} + \nu \operatorname{grad} \mathbf{u}) &= \bar{\mathbf{s}} && \text{in } \Omega \times (0, T], \\
 \operatorname{div} \mathbf{u} &= 0 && \text{in } \Omega \times (0, T], \\
 \mathbf{u} &= \bar{\mathbf{u}} && \text{on } \partial\Omega_D \times (0, T], \\
 (-p\mathbf{I} + \nu \operatorname{grad} \mathbf{u}) \mathbf{n} &= \bar{\mathbf{t}} && \text{on } \partial\Omega_N \times (0, T], \\
 \mathbf{u} &= \mathbf{u}_0 && \text{in } \Omega \times \{t = 0\},
 \end{aligned} \tag{2.23}$$

where \mathbf{u}_0 is the initial velocity field in the domain and T is the total time considered. The transient term is discretized using Backward Differentiation Formulae (BDF) schemes that belongs to the family of multi-step linear methods. They are discussed briefly in the following sections.

2.8.2 Backward Differentiation Formulae (BDF) schemes

The time derivative term can be discretized using BDF schemes (Gear, 1971, Brenan *et al.*, 1995) as follows,

$$\frac{\partial \mathbf{u}}{\partial t} \approx \frac{\alpha_\mu \mathbf{u}_{n+1} - \mathbf{u}_{n,\text{BDF}\mu}}{\Delta t}, \tag{2.24}$$

where Δt is the time step, \mathbf{u}_{n+1} is the velocity at the time instant t_{n+1} . The constant α_μ and the term $\mathbf{u}_{n,\text{BDF}\mu}$ depends on the order μ of the BDF scheme. In the present work, BDF schemes of order 1, 2 and 3 are used. The parameters in equation (2.24) are defined as follows,

$$\mathbf{u}_{n,\text{BDF}\mu} = \begin{cases} \mathbf{u}_n & \text{if } n \geq 0 \text{ for } \mu = 1 \text{ (BDF1),} \\ 2\mathbf{u}_n - \frac{1}{2}\mathbf{u}_{n-1} & \text{if } n \geq 1 \text{ for } \mu = 2 \text{ (BDF2),} \\ 3\mathbf{u}_n - \frac{3}{2}\mathbf{u}_{n-1} + \frac{1}{3}\mathbf{u}_{n-2} & \text{if } n \geq 2 \text{ for } \mu = 3 \text{ (BDF3),} \end{cases} \tag{2.25}$$

Table 2.5: Values of $\beta_{\mu,i}$, $i = 1, \dots, \mu$ and α_μ for variable step BDF schemes.

	$\mu = 1$	$\mu = 2$	$\mu = 3$
α_μ	1	$\frac{2\omega_{n+1} + 1}{\omega_{n+1} + 1}$	$\frac{\omega_{n+1}\omega_n}{\omega_{n+1}\omega_n + \omega_n + 1} + \frac{2\omega_{n+1} + 1}{\omega_{n+1} + 1}$
$\beta_{\mu,1}$	1	$(\omega_{n+1} + 1)$	$\frac{(\omega_{n+1} + 1)(\omega_{n+1}\omega_n + \omega_n + 1)}{\omega_n + 1}$
$\beta_{\mu,2}$	—	$-\frac{\omega_{n+1}^2}{1 + \omega_{n+1}}$	$-\frac{\omega_n^2(\omega_{n+1}\omega_n + \omega_n + 1)}{\omega_{n+1} + 1}$
$\beta_{\mu,3}$	—	—	$\frac{(\omega_{n+1} + 1)\omega_{n+1}^2\omega_n^3}{(\omega_n + 1)(\omega_{n+1}\omega_n + \omega_n + 1)}$

and

$$\alpha_\mu = \begin{cases} 1 & \text{for } \mu = 1 \text{ (BDF1),} \\ \frac{3}{2} & \text{for } \mu = 2 \text{ (BDF2),} \\ \frac{11}{6} & \text{for } \mu = 3 \text{ (BDF3).} \end{cases} \quad (2.26)$$

The above notation is adopted from the work of Forti and Dedè (2015). This is the standard BDF schemes for constant time steps. In the case of variable time steps the equation (2.25) can be rewritten as,

$$\mathbf{u}_{n,\text{BDF}\mu} = \begin{cases} \beta_{\mu,1}\mathbf{u}_n & \text{if } n \geq 0 \text{ for } \mu = 1 \text{ (BDF1),} \\ \beta_{\mu,1}\mathbf{u}_n - \beta_{\mu,2}\mathbf{u}_{n-1} & \text{if } n \geq 1 \text{ for } \mu = 2 \text{ (BDF2),} \\ \beta_{\mu,1}\mathbf{u}_n - \beta_{\mu,2}\mathbf{u}_{n-1} + \beta_{\mu,3}\mathbf{u}_{n-2} & \text{if } n \geq 2 \text{ for } \mu = 3 \text{ (BDF3),} \end{cases} \quad (2.27)$$

where the constants depends on the ratios of successive time steps which are defined as follows,

$$\omega_{n+1} = \frac{\Delta t_{n+1}}{\Delta t_n}, \quad \omega_n = \frac{\Delta t_n}{\Delta t_{n-1}}. \quad (2.28)$$

Using the definitions in equation (2.28), the constants in equations (2.27) are presented in table 2.5.

The necessary and sufficient condition for a linear multi-step to be convergent is if it is both consistent and zero-stable (Lambert, 1991). The scheme is consistent if the order conditions stated in Lambert (1991) are satisfied. The coefficients of the variable step BDF scheme readily satisfies the order conditions and hence, it is only needed to check for zero-stability.

Zero-stability quantifies the propagation computational errors that come from the perturbations as the time step approaches zero. Thus, a zero-stable multi-step method is insensitive to perturbations such as round-off errors. Performing this analysis for the variable step BDF schemes impose a restriction on the ratio of the time steps, ω_{n+1} and ω_n . This analysis is presented in Grigorieff (1983) and Wang and Ruuth (2008) in the context of variable step BDF scheme and variable step implicit-explicit linear multi-step methods (VSIMEX), respectively. In the case of variable step BDF2 scheme, the method is zero-stable if $\omega_{n+1} \leq 1 + \sqrt{2}$. In the case of variable step BDF3 scheme, the analysis is complicated because of involvement of two time step ratios instead of one as in the case of variable step BDF2 scheme. The bounds reported by several works for variable step BDF3 scheme are presented in table 2.6. As stated

Table 2.6: Upper bounds for variable step BDF3 scheme. The scheme is zero-stable if $\omega_{n+1} < R$.

Reference	R_b
Wang and Ruuth (2008)	1.1273
Grigorieff (1983)	1.127
Calvo <i>et al.</i> (1990)	1.476
Guglielmi and Zennaro (2001)	1.501

in Grigorieff (1983), these limits can be unrealistic depending on the type of problem being solved. However, some of the numerical examples presented in Wang and Ruuth (2008), violates these limits without loss of stability of the method.

Using the definitions in equations (2.24) and (2.27), the weak form of unsteady incompressible Navier–Stokes equations using CG can be stated as: find $\mathbf{u}_{n+1} \in [\mathcal{H}^1(\Omega)]^2$ and $p_{n+1} \in \mathcal{L}_2(\Omega)$ such that $\mathbf{u}_{n+1} = \Pi^h(\bar{\mathbf{u}}_{n+1})$ on $\partial\Omega_D$, $\mathbf{u} = \mathbf{u}_0$ at $t = 0$ and,

$$\begin{aligned}
& \left(\delta \mathbf{u}, \frac{\alpha_\mu \mathbf{u}_{n+1}}{\Delta t} \right)_\Omega + (\delta \mathbf{u}, (\text{grad } \mathbf{u}_{n+1}) \mathbf{u}_{n+1})_\Omega + (\text{grad } \delta \mathbf{u}, (-p_{n+1} \mathbf{I} + \nu \text{grad } \mathbf{u}_{n+1}))_\Omega \\
& - (\delta \mathbf{u}, \bar{\mathbf{s}}_{n+1})_\Omega - \langle \delta \mathbf{u}, \bar{\mathbf{t}}_{n+1} \rangle_{\partial\Omega_N} - \left(\delta \mathbf{u}, \frac{\mathbf{u}_{n,\text{BDF}\mu}}{\Delta t} \right)_\Omega = 0, \\
& - (\delta p, \text{div } \mathbf{u}_{n+1})_\Omega = 0,
\end{aligned} \tag{2.29a}$$

for all $\delta \mathbf{u} \in [\mathcal{H}^1(\Omega)]^2$, $\delta p \in \mathcal{L}_2(\Omega)$ such that $\delta \mathbf{u} = \mathbf{0}$ on $\partial\Omega_D$.

The HDG weak form with the BDF time discretization can be stated as: find $(\mathbf{u}_{n+1}, p_{n+1}, \mathbf{L}_{n+1}, \hat{\mathbf{u}}_{n+1}, \rho_{n+1,e}) \in [\mathcal{V}_k^h(\Omega)]^2 \times [\mathcal{V}_k^h(\Omega)] \times [\mathcal{V}_k^h(\Omega)]^{2 \times 2} \times [\Lambda_k^h(\Gamma)]^2 \times \mathbb{R}^{n_{el}}$ in every element Ω^e such that $\mathbf{u} = \mathbf{u}_0$ at $t = 0$ and,

$$\begin{aligned}
& (\delta \mathbf{L}, \mathbf{L}_{n+1})_{\Omega^e} + (\text{div } \delta \mathbf{L}, \mathbf{u}_{n+1})_{\Omega^e} - \langle \delta \mathbf{L} \mathbf{n}, \hat{\mathbf{u}}_{n+1} \rangle_{\partial\Omega^e} = 0, \\
& \left(\delta \mathbf{u}, \frac{\alpha_\mu \mathbf{u}_{n+1}}{\Delta t} \right)_{\Omega^e} - (\text{grad } \delta \mathbf{u}, \mathbf{u}_{n+1} \otimes \mathbf{u}_{n+1})_{\Omega^e} - (\delta \mathbf{u}, \text{div } (-p_{n+1} \mathbf{I} + \nu \mathbf{L}_{n+1}))_{\Omega^e} \\
& + \langle \delta \mathbf{u}, (\hat{\mathbf{u}}_{n+1} \otimes \hat{\mathbf{u}}_{n+1}) \mathbf{n} + \tau_u (\mathbf{u}_{n+1} - \hat{\mathbf{u}}_{n+1}) \rangle_{\partial\Omega^e} \\
& - (\delta \mathbf{u}, \bar{\mathbf{s}}_{n+1})_{\Omega^e} - \left(\delta \mathbf{u}, \frac{\mathbf{u}_{n,\text{BDF}\mu}}{\Delta t} \right)_{\Omega^e} = 0,
\end{aligned} \tag{2.30a}$$

$$- (\text{grad } \delta p, \mathbf{u}_{n+1})_{\Omega^e} + \langle \delta p, \hat{\mathbf{u}}_{n+1} \cdot \mathbf{n} \rangle_{\partial\Omega^e} = 0,$$

$$\frac{1}{|\partial\Omega^e|} \langle p_{n+1}, 1 \rangle_{\partial\Omega^e} = \rho_{n+1,e} \text{ for } e = 1, \dots, n_{el},$$

$$\sum_{e=1}^{n_{el}} \langle \delta \hat{\mathbf{u}}, (-p_{n+1} \mathbf{I} + \nu \mathbf{L}_{n+1}) \mathbf{n} + \tau_u (\hat{\mathbf{u}}_{n+1} - \mathbf{u}_{n+1}) \rangle_{\partial\Omega^e} = \langle \delta \hat{\mathbf{u}}, \bar{\mathbf{t}}_{n+1} \rangle_{\partial\Omega_N}, \tag{2.30b}$$

$$\langle \hat{\mathbf{u}}_{n+1} \cdot \mathbf{n}, 1 \rangle_{\partial\Omega^e} = 0 \text{ for } e = 1, \dots, n_{el},$$

$$\hat{\mathbf{u}}_{n+1} = \mathbb{P}_2(\bar{\mathbf{u}}_{n+1}) \text{ on } \partial\Omega_D,$$

for all $(\delta \mathbf{u}, \delta p, \delta \mathbf{L}, \delta \hat{\mathbf{u}}) \in [\mathcal{V}_k^h(\Omega)]^2 \times [\mathcal{V}_k^h(\Omega)] \times [\mathcal{V}_k^h(\Omega)]^{2 \times 2} \times [\Lambda_k^h(\Gamma)]^2$, such that $\delta \hat{\mathbf{u}} = \mathbf{0}$ on $\partial\Omega_D$. The discrete system is similar to the one obtained in the steady state case except for the new terms arising from time derivative. The same solution procedure described earlier can be applied to the unsteady case at each time step. It is also possible to apply the local post-processing to get the super-convergent solution at any desired time step.

First order BDF (BDF1) scheme coincides with the backward Euler scheme and it is self-starting as it needs information only from the immediate prior time step. In the case of other BDF schemes, if μ is the order of the method, it needs first $\mu - 1$ solution values for the method to start and it is evident from equation (2.25). In the present work, it is done by using BDF1 for the first time step and, BDF2 for second and third time steps. This procedure does not guarantee third order accuracy for BDF3 scheme. But, BDF3 is only used as error estimator in adaptive time stepping scheme and only convergence results of BDF1 and BDF2 schemes are presented.

2.8.3 Taylor–Vortex flow

The benchmark example for unsteady Navier–Stokes equations considered in Maday *et al.* (1990) is chosen for convergence analysis. The exact solution for Taylor–Vortex problem is

$$\begin{aligned} \mathbf{u} &= \begin{bmatrix} -\cos(\pi x_1) \sin(\pi x_2) \exp\left(\frac{-2\pi^2 t}{Re}\right) \\ \sin(\pi x_1) \cos(\pi x_2) \exp\left(\frac{-2\pi^2 t}{Re}\right) \end{bmatrix}, \\ p &= -\frac{1}{4} (\cos(2\pi x_1) + \cos(2\pi x_2)) \exp\left(\frac{-4\pi^2 t}{Re}\right) + C, \end{aligned} \quad (2.31)$$

where $Re = 1/\nu$ is the Reynolds number and C is a constant. The problem is considered in the domain $[0, 1] \times [0, 1]$ with Reynolds number $Re = 20$. Dirichlet boundary conditions are prescribed on all the exterior boundary to be equal to the analytical solution and final time $T = 1$ is considered. The analytical solution at the end of $T = 1$ is shown in the fig. 2.24. The initial condition is provided from the analytical solution at $t = 0$. The solution satisfies the Navier–Stokes equations without any body force and hence, prescribed body force is zero.

A uniform triangular mesh with $k = 4$ and $h = 1/2^5$ along with the stabilisation parameter, $\tau_u = 1$, is used on all faces in the results presented for this example. The mesh considered here is fine enough to keep the spatial errors very low and hence, the dominant errors are the temporal ones. Time steps of $1/2, 1/2^2, 1/2^3$ and $1/2^4$ are used to plot the convergence of BDF1 and BDF2 schemes.

Figure 2.25 shows the \mathcal{L}_2 norm of the error in the post processed velocity of HDG and the error in velocity of CG. Similarly, fig. 2.26a presents the \mathcal{L}_2 norm of the error in pressure for HDG and CG. It can be noticed that both time stepping schemes display the theoretical rate of convergence for both velocity and pressure. Since BDF2 scheme is not self starting, the first step is computed using BDF1 scheme and the rest of the steps using BDF2. As Dirichlet boundary conditions are prescribed on all boundary, the pressure is prescribed to zero at one node in CG where as in case of HDG, the mean of the pressure is prescribed to zero in one element. The constant of the pressure is computed by taking the difference between the analytical pressure and the numerical value at a node in the mesh. The errors in pressure are computed using the obtained constant value in both CG and HDG cases. It can be noticed that the errors in both velocity and pressure for each corresponding time step of HDG and CG are almost identical and have similar rates of convergence as spatial discretization errors in this case are very small and errors from time discretization dominate.

Convergence analysis in space is also presented in the context of time discretization schemes. A time step of 2×10^{-4} is used to keep the temporal errors small enough that cannot influence the convergence rates in space. BDF2 scheme is used for time integration while degrees of approximation $k = 2 - 5$ are used. The stabilisation parameter, $\tau_u = 1$, is used on all faces and error at the final time instance, $T = 1$ sec is computed.

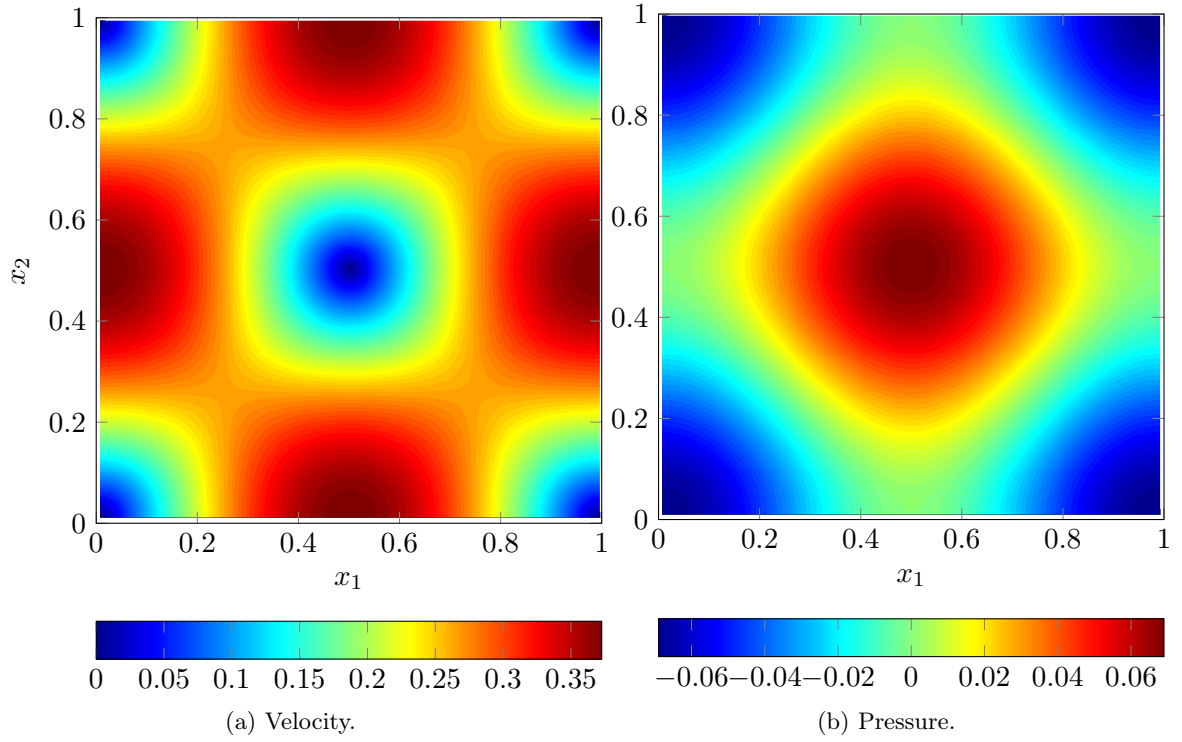


Figure 2.24: Taylor-Vortex flow: analytical solution.

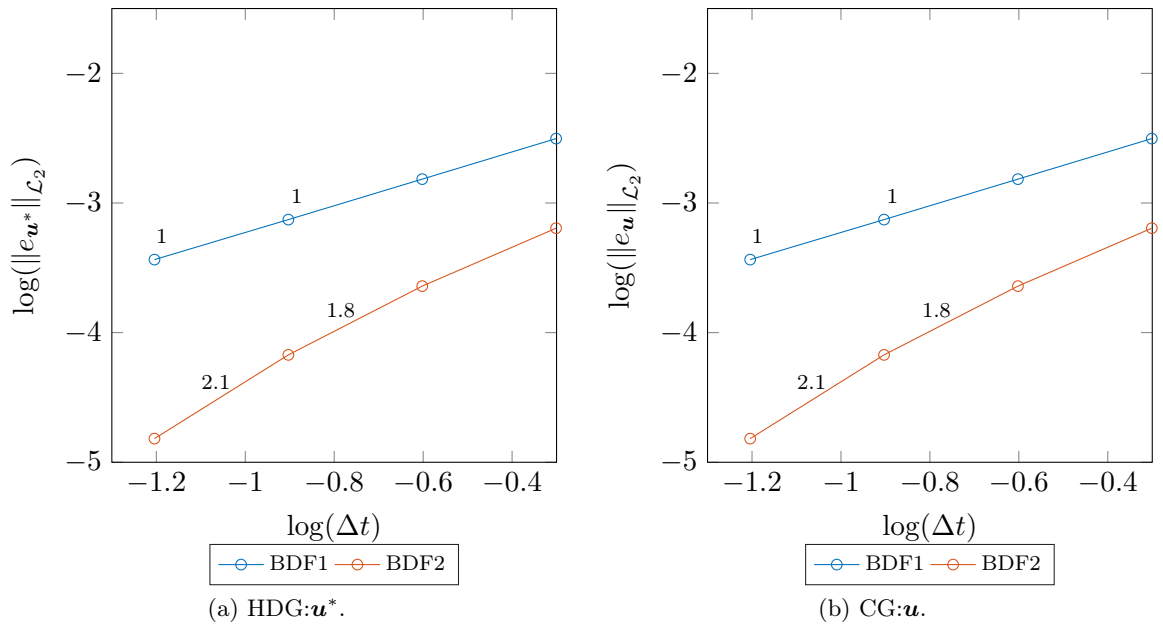


Figure 2.25: Taylor-Vortex flow: convergence of velocity in time.

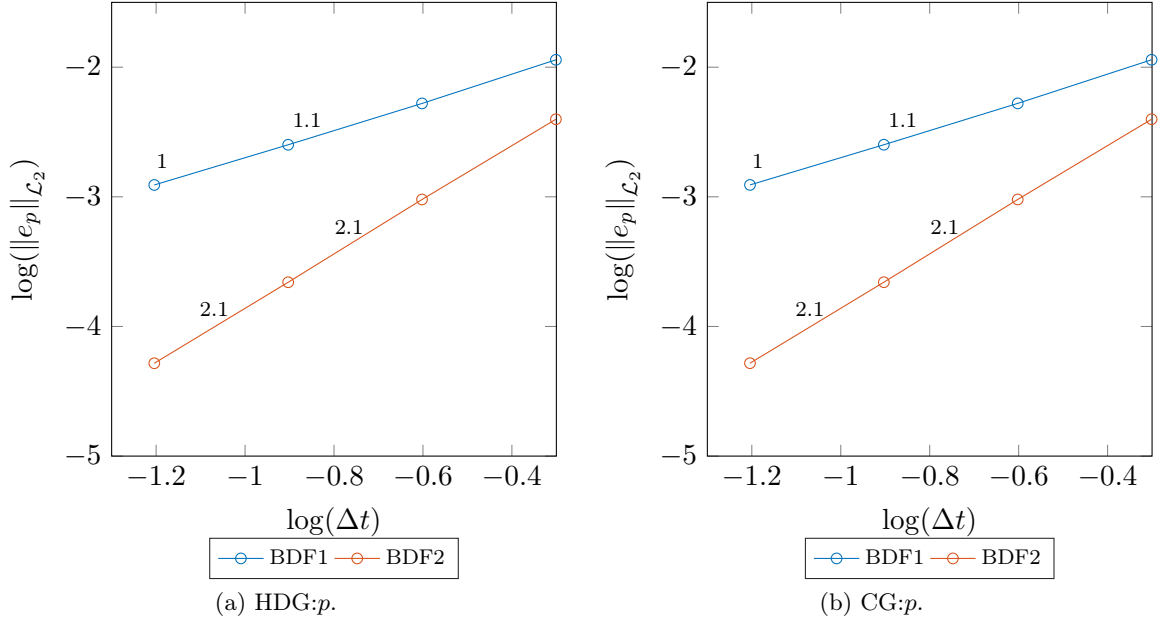


Figure 2.26: Taylor–Vortex flow: convergence of pressure in time.

Table 2.7: Partitioning scheme for the time interval $[0, 2]$.

Time interval	$[0, 0.4]$	$[0.4, 0.8]$	$[0.8, 1.2]$	$[1.2, 1.6]$	$[1.6, 2.0]$
Number of time steps	16	14	6	6	8

Figures 2.27 and 2.28 shows the convergence plots for velocity and pressure, respectively. It can be observed that the optimal convergence rates in space are obtained in both CG and HDG discretizations. HDG produces smaller error for same degree and mesh for both velocity and pressure as noticed in earlier example of Kovasznay flow in Section 2.6. This example concludes a complete convergence analysis in space and time for Taylor–Vortex flow.

Finally, a convergence analysis is presented for the case of variable step BDF schemes. To perform the analysis, the procedure presented in the work of Wang and Ruuth (2008) is adopted here. The final time, T , is taken as 2 seconds and the whole time interval, $[0, 2]$, is divided into 5 sub-intervals of equal length. Inside each sub-interval, a different time step is taken and hence, a variable time step scheme is obtained. Table 2.7 presents the number of time steps used in each sub-interval for the present analysis. The scheme presented in table 2.7 is the coarsest temporal grid with a total number of time steps of 50. Four different temporal grids are obtained by successively doubling the number of steps in each sub-interval. Therefore, the total number of time steps presented in the analysis are $\{50, 100, 200, 400\}$. The same spatial mesh of $k = 4$ and $h = 1/2^5$ as the previous example is chosen for the analysis.

The convergence plots are presented for BDF1, BDF2 and BDF3 schemes with post-processed velocity of HDG and velocity of CG in fig. 2.29. It can be noticed that the optimal convergence rates are obtained for all the three schemes used in the analysis. Even though, BDF3 scheme needs the first step to be started with a third-order accurate method to recover the optimal rate, in the present case, the partitioned scheme is chosen in such a way that the errors from the first few time steps do not influence the global error. The global errors in

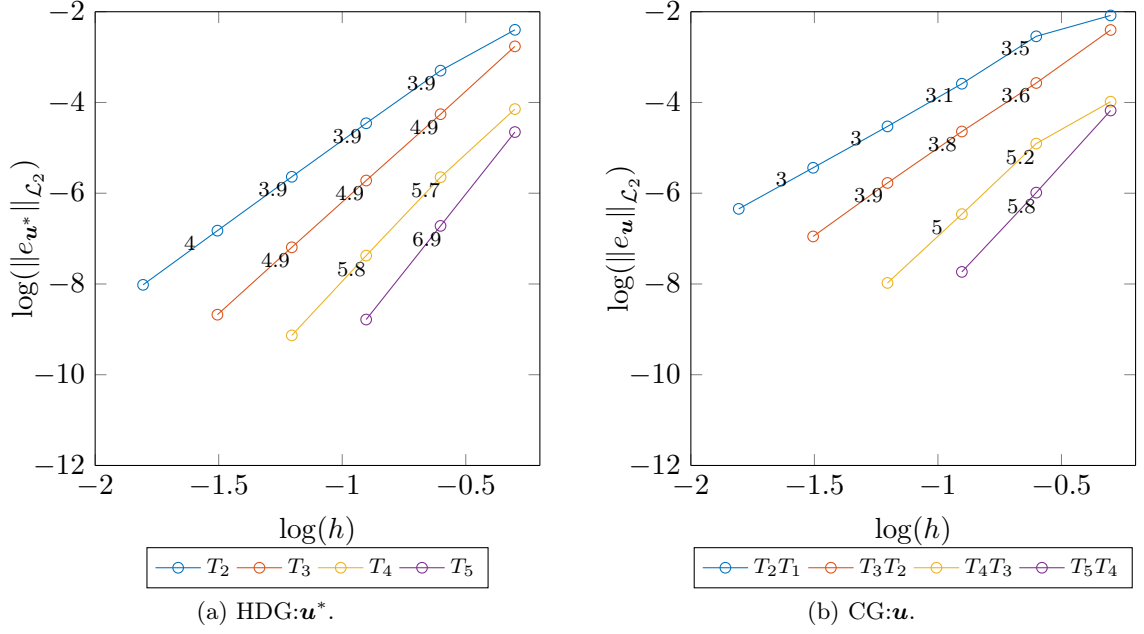


Figure 2.27: Taylor-Vortex flow: convergence of velocity in space.

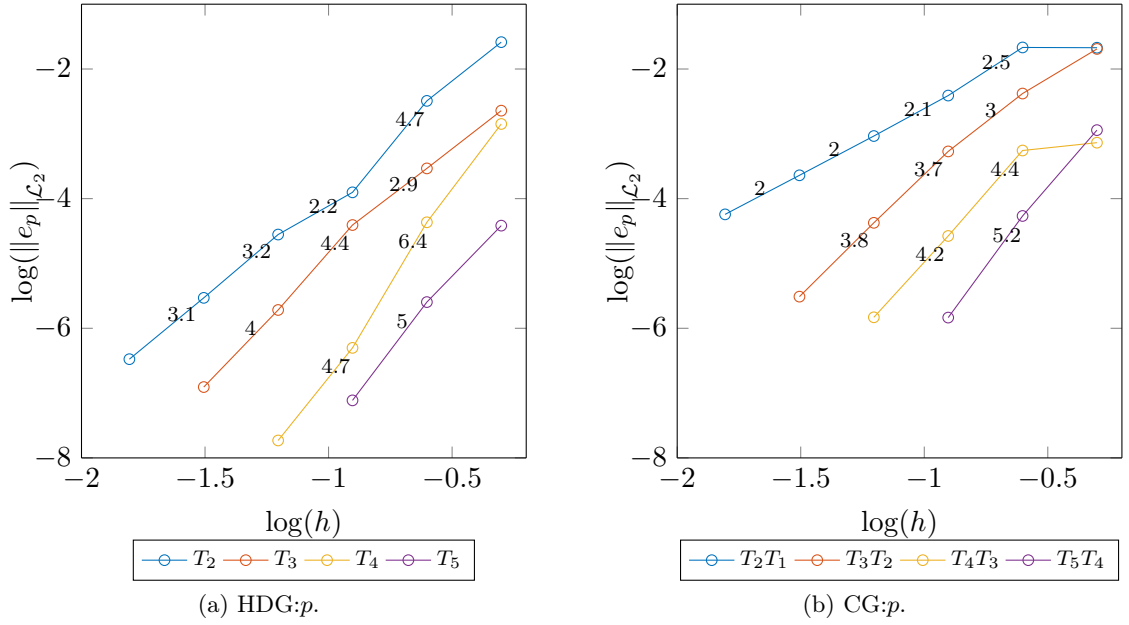


Figure 2.28: Taylor-Vortex flow: convergence of pressure in space.

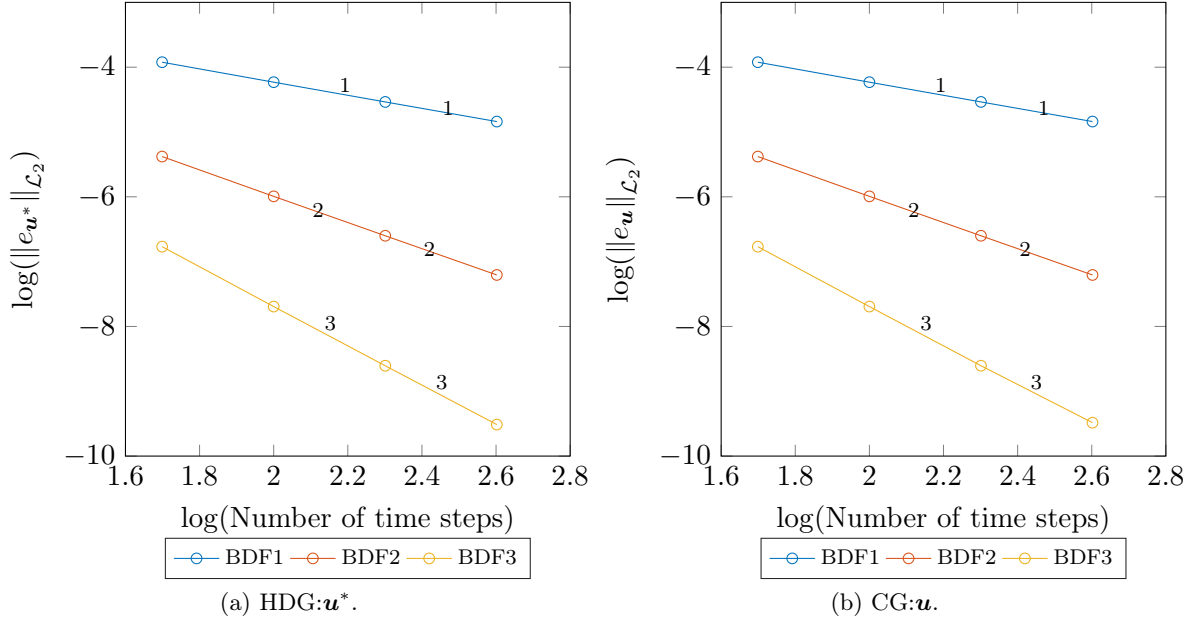


Figure 2.29: Taylor–Vortex flow: convergence of velocity for variable time stepping scheme.

both CG and HDG are almost identical confirming that the spatial errors are negligible and temporal errors are dominant.

2.8.4 Error estimation and time step adaption

In this section, a standard technique for estimating the local error in each time step using BDF3 scheme is considered. The local truncation errors of BDF schemes can be represented as follows,

$$\begin{aligned} \|\mathbf{u}_{n,a} - \mathbf{u}_{n,1}\| &= c_1 \Delta t^2 + \mathcal{O}(\Delta t^3), \\ \|\mathbf{u}_{n,a} - \mathbf{u}_{n,2}\| &= c_2 \Delta t^3 + \mathcal{O}(\Delta t^4), \\ \|\mathbf{u}_{n,a} - \mathbf{u}_{n,3}\| &= c_3 \Delta t^4 + \mathcal{O}(\Delta t^5), \end{aligned} \quad (2.32)$$

where $\mathbf{u}_{n,a}$ is the analytical solution and $\mathbf{u}_{n,\mu}$ is the numerical solution of BDF scheme of order μ at n^{th} time step. Therefore, for sufficiently small time step, Δt , the estimation of local discretization errors for BDF1 and BDF2 using the solution of BDF3 scheme can be shown as,

$$\begin{aligned} e_{n,1} &= \|\mathbf{u}_{n,3} - \mathbf{u}_{n,1}\| \approx c_1 \Delta t^2, \\ e_{n,2} &= \|\mathbf{u}_{n,3} - \mathbf{u}_{n,2}\| \approx c_2 \Delta t^3. \end{aligned} \quad (2.33)$$

Suppose, the local error in the new time step should be the order of a given tolerance, δ_{tol} , i.e., it is required that $e_{n,1} < \delta_{tol}$ or $e_{n,2} < \delta_{tol}$. Hence, a new time step can be chosen in such a way that,

$$e_{n,1} \approx \delta_{tol}, \quad e_{n,2} \approx \delta_{tol}, \quad (2.34)$$

and the optimal new time step, $\bar{\Delta t}$, satisfies,

$$\delta_{tol} \approx c_{13} \bar{\Delta t}^2, \quad \delta_{tol} \approx c_{23} \bar{\Delta t}^3. \quad (2.35)$$

From equations (2.33) and (2.35), it can be shown that,

$$\bar{\Delta t} = \Delta t \left(\frac{\delta_{tol}}{e_{n,\mu}} \right)^{\frac{1}{\mu+1}} \quad \text{for } \mu = 1, 2. \quad (2.36)$$

The equation (2.36) gives the optimal time step at the n^{th} level to have the local error within δ_{tol} . In practice, it is not efficient to repeat every time step with the new optimal length, if the difference between $\bar{\Delta t}$ and Δt is not significantly large. Hence, this new optimal time step is used for the next time step level in the implementation. Hence, equation (2.36) can be rewritten as,

$$\Delta t_{n+1} = \Delta t_n \left(\frac{\delta_{tol}}{e_{n,\mu}} \right)^{\frac{1}{\mu+1}} \quad \text{for } \mu = 1, 2. \quad (2.37)$$

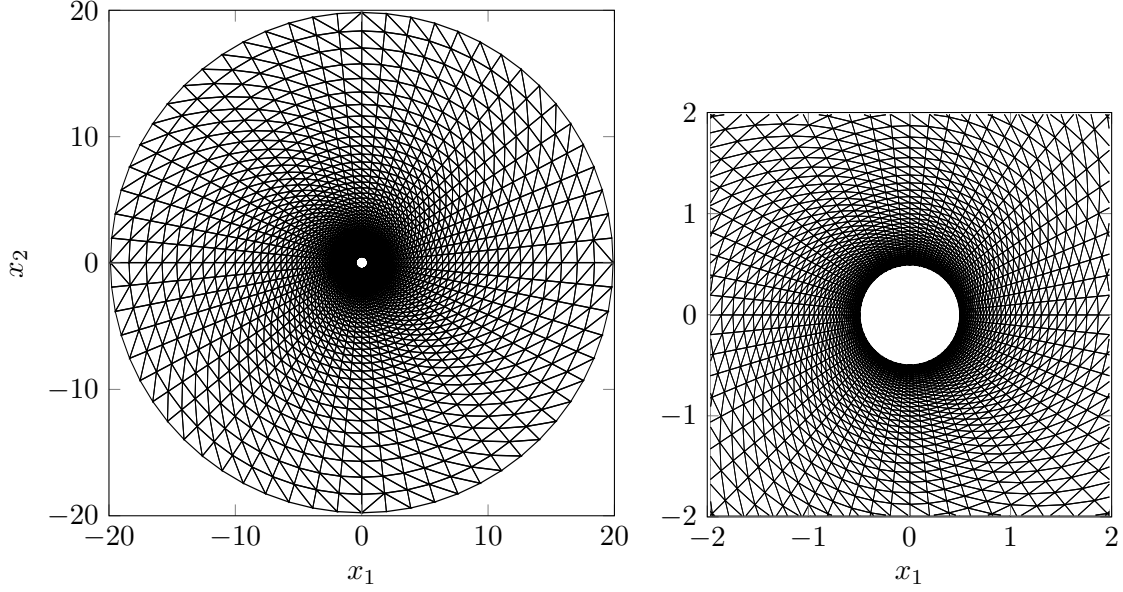
Owing to the limitations of the ratio of time steps discussed before in the variable step BDF schemes, the proposed algorithm for adaptive time step is presented in algorithm 1. The

Algorithm 1: Adaptive time step algorithm for BDF1 and BDF2 schemes.

Input : $\delta_{tol} > 0, \mu = \{1, 2\}, \Delta t_n > 0, \mathbf{u}_{n,\text{BDF}\mu}$
Output: Δt_{n+1}

- 1 Compute $\mathbf{u}_{n+1,\mu}$ and $\mathbf{u}_{n+1,3}$ using BDF scheme of order μ and BDF3 schemes, respectively ;
- 2 Compute $e_{n,\mu}$;
- 3 Compute $\bar{\Delta t}$ using equation (2.36) ;
- 4 **if** $\frac{\bar{\Delta t}_n}{\Delta t_n} \geq b_2$ **then**
 - 5 $\Delta t_{n+1} = \min(\bar{\Delta t}_n, b_3 \Delta t_n)$;
 - 6 Set $\mathbf{u}_{n+1} = \mathbf{u}_{n+1,3}$;
 - 7 Put $n = n + 1$;
 - 8 Go to next time step;
- 9 **else if** $\frac{\bar{\Delta t}_n}{\Delta t_n} \leq b_1$ **then**
 - 10 $\Delta t_n = \bar{\Delta t}_n$;
 - 11 Repeat the time step;
- 12 **else**
 - 13 $\Delta t_{n+1} = \Delta t_n$;
 - 14 Go to next time step;

constants b_1 , b_2 and b_3 depend on the limitations of the time step ratios discussed earlier. In the present work, $b_1 = 0.8$ and $b_2 = 1.1$ and $b_3 = 1.5$ are used. The algorithm does not change the time step if the ratio $\frac{\bar{\Delta t}_n}{\Delta t_n}$ is in the interval $[b_1, b_2]$. This is to avoid small changes in the time step and hence, less variations. The constant b_3 is a direct consequence of the limitation of maximum ratio from the zero-stability analysis. In the present algorithm, since, BDF3 solution is computed at each time step, it is taken as the solution at that time step. Hence, the maximum time step ratio corresponding to BDF3 scheme is chosen to have a stable adaptive algorithm.



(a) Circular computational mesh.

(b) Mesh around cylinder.

Figure 2.30: Flow around cylinder: computational mesh and zoom.

2.8.5 Flow around the cylinder

In this section, the results of 2-D flow around the stationary cylinder are presented for HDG and CG using constant step and variable time step BDF schemes. A circular cylinder of radius 0.5 is considered in a circular domain of radius 20. The mesh used to perform the analysis is shown in fig. 2.30. The degree of approximation, $k = 4$ is used in all the analysis for the present example. The left half of the exterior boundary of the computational domain is inflow and a constant velocity of $1\mathbf{e}_1$ is prescribed. On the outflow boundary, homogeneous Neumann boundary condition is specified. No-slip conditions are applied on the boundary of the cylinder. The kinematic viscosity, ν , is taken as 0.01 and it corresponds to a Reynolds number of 100. A total time, T , of 160 seconds is considered in the analysis. A velocity of $1\mathbf{e}_1$ is prescribed everywhere in the domain at time, $t = 0$. In the case of constant time step, $\Delta t = 0.05$ is used in both CG and HDG analysis. Lift and drag coefficients are used for presenting the results. They are defined as follows,

$$C_l = \frac{F_l}{0.5\rho \|\mathbf{u}\|_\infty^2 D}, \quad \text{and} \quad C_d = \frac{F_d}{0.5\rho \|\mathbf{u}\|_\infty^2 D}, \quad (2.38)$$

where ρ is the density, D is the diameter of the cylinder, which is taken as unity here and $\|\mathbf{u}\|_\infty$ is the magnitude of the flow velocity, which is 1 in present example. F_l and F_d are the lift and drag forces acting on the cylinder per unit area and they are computed using following expressions,

$$F_l = \int_S \rho (-p n_2 - \mathbf{t} \cdot (\mathbf{e}_3 \times \mathbf{n}) n_1) \, dS, \quad \text{and} \quad F_d = \int_S \rho (-p n_1 + \mathbf{t} \cdot (\mathbf{e}_3 \times \mathbf{n}) n_2) \, dS \quad (2.39)$$

where $\mathbf{t} \cdot (\mathbf{e}_3 \times \mathbf{n})$ is the tangential shear component of the viscous stress vector $\mathbf{t} = \nu(\text{grad } \mathbf{u} + (\text{grad } \mathbf{u})^T)\mathbf{n}$ and \mathbf{n} is unit normal vector to the surface S , which can be written as $(n_1\mathbf{e}_1 +$

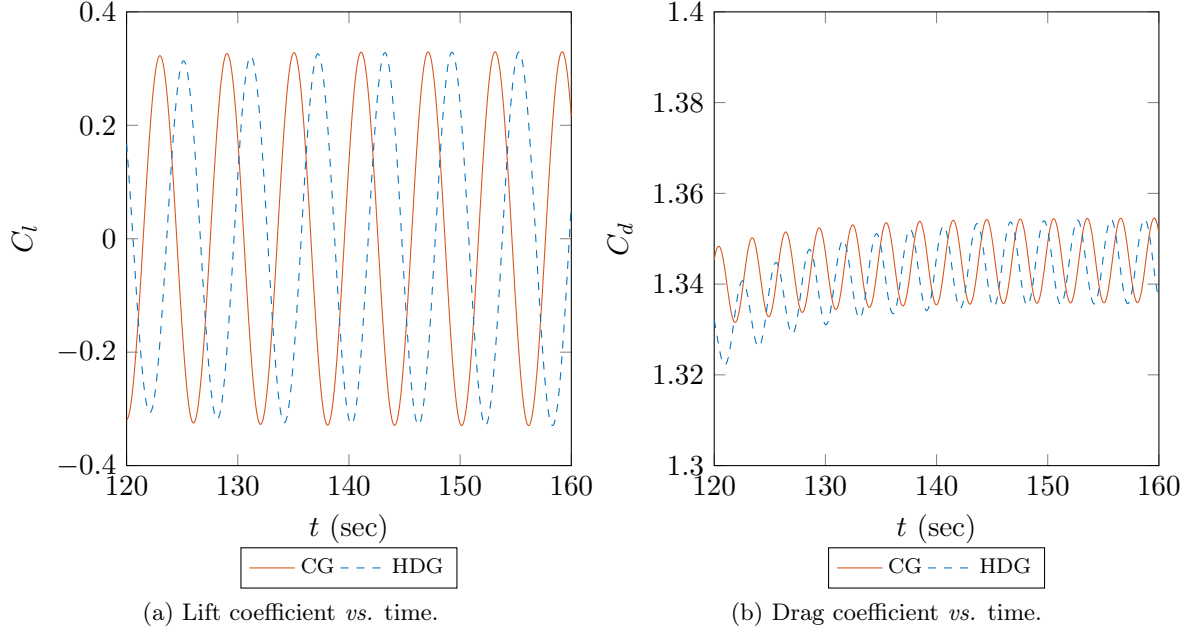


Figure 2.31: Flow around cylinder: variation of lift and drag coefficients with time for constant time step of 0.05 sec using HDG and CG.

$n_2 \mathbf{e}_2$). The quantity is $(\mathbf{e}_3 \times \mathbf{n})$ is a tangent vector to surface S and it can be expressed as $(n_2 \mathbf{e}_1 - n_1 \mathbf{e}_2)$.

Figure 2.31 shows the variation of lift and drag coefficients with time with both HDG and CG discretizations with constant time step of 0.05 sec. It is well documented in the literature (Rajani *et al.*, 2009) that the flow becomes destabilized around the $Re = 40 - 50$, generating the asymmetric eddy patterns which are well-known as von Kármán vortex street. The present results show that the flow first reaches a steady state. This stabilised flow is destabilised by the perturbation effects which comes from the numerical approximation schemes, truncation and round-off errors (Braza *et al.*, 1986). Figure 2.32 shows the velocity contours at two different time instances, *i.e.*, fig. 2.32a is in steady state and fig. 2.32b is in vortex shedding regime. In the fig. 2.31, the lift and drag coefficients are out of phase between CG and HDG owing to different time instances when flow destabilises. Another quantity of interest in this example is Strouhal number (St). Strouhal number is a non-dimensional number used as a measure of the oscillating flow phenomenon in the wake of the cylinder. It is defined as,

$$St = \frac{f_s D}{\|\mathbf{u}\|_\infty}, \quad (2.40)$$

where f_s is the shedding frequency, D is the diameter of the cylinder and $\|\mathbf{u}\|_\infty$ is the magnitude of the flow velocity. The shedding frequency in the present work is computed using FFT analysis. Table 2.8 compares the lift, drag coefficients and Strouhal number with present results and literature ones. It is evident from the table 2.8 that the present results are very close to literature results and also, HDG and CG give very similar results. Note that the lift and drag coefficients are computed only for the last 10 seconds of the flow where both HDG and CG have reached an unsteady oscillatory regime.

Now, the problem is solved with adaptive time stepping scheme with HDG and CG dis-

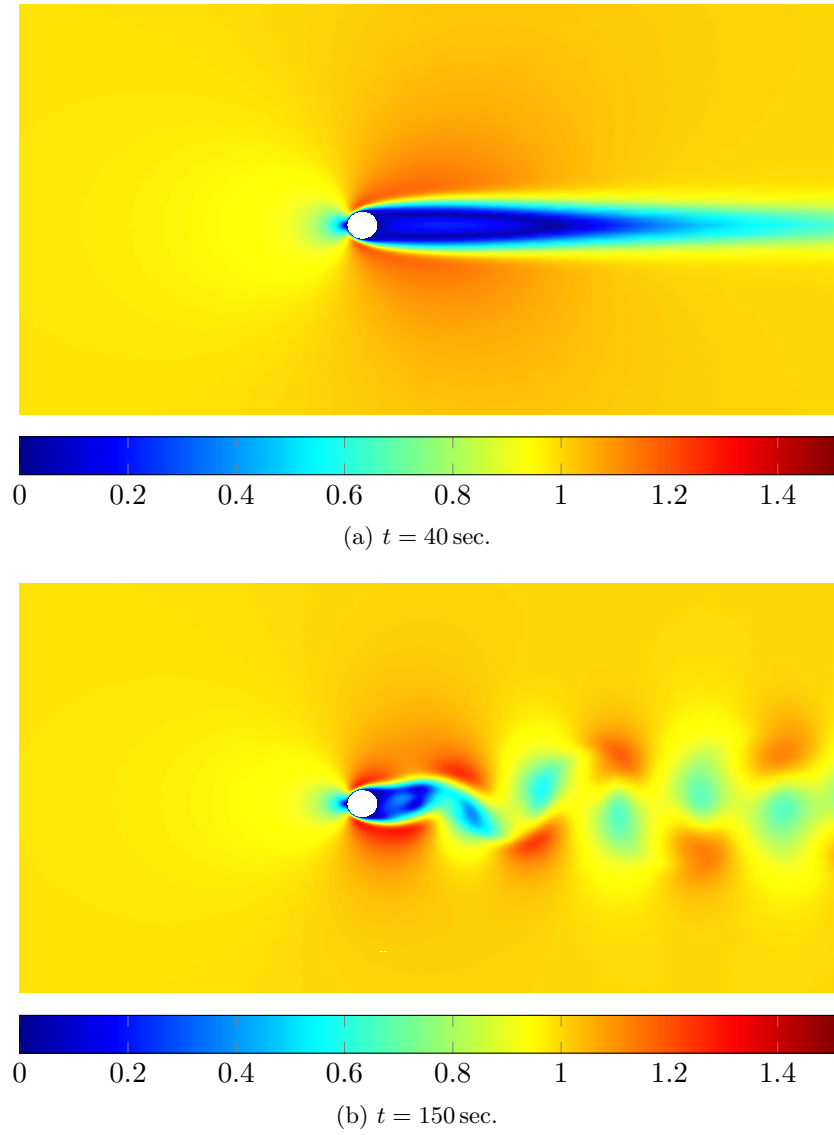


Figure 2.32: Flow around cylinder: velocity contours at different time instances.

Table 2.8: Lift, drag coefficients and Strouhal number for $Re = 100$ with constant time step of 0.05 sec.

Variable	Braza <i>et al.</i> (1986)	Liu <i>et al.</i> (1998)	Present CG/HDG
Lift (C_l)	0 ± 0.25	0 ± 0.339	0 ± 0.329
Drag (C_d)	1.364 ± 0.015	1.350 ± 0.012	1.345 ± 0.009
Strouhal number (St)	0.160	0.164	0.164

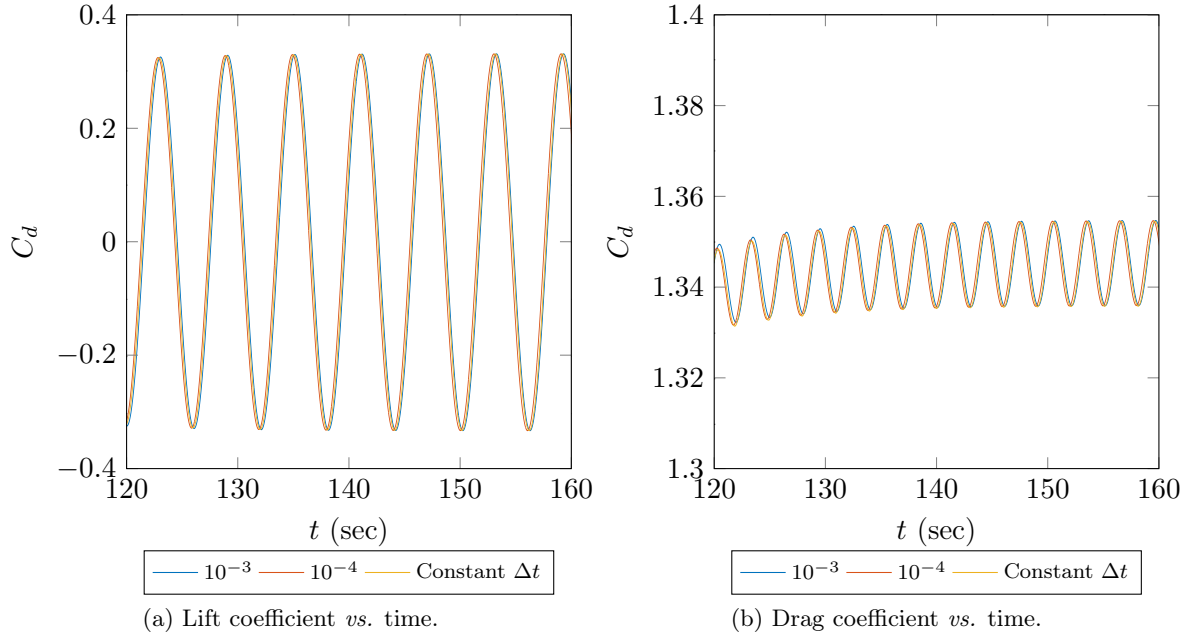


Figure 2.33: Flow around cylinder: variation of lift and drag coefficients with time for adaptive time stepping scheme for different levels of local error for CG.

cretizations. Two different levels of local error, δ_{tol} , are considered at 10^{-3} and 10^{-4} . In this case, the simulations are carried out with a constant time step of 0.05 for the first 20 seconds and then adaptive scheme is employed. This ensures that all the non-physical oscillations that can occur due to initial data will die before adaption scheme is used. Figs. 2.33 and 2.34 shows the variation of lift and drag coefficients for different local errors specified using CG and HDG discretizations, respectively. It can be observed from the figs. 2.33 and 2.34 that the variation of lift and drag coefficients with adaptive time stepping scheme is very similar to the ones with constant time step of 0.05 sec. Hence, it can be concluded that the chosen time step of 0.05 sec for constant time step analysis is already small enough to get a time accurate solution. The variation of time step with flow time is shown in fig. 2.35 for both CG and HDG discretizations for two levels of local error tolerances. There are few important conclusions that can be drawn from these plots. Firstly, when the specified local error tolerance is lower, the maximum time step that the scheme adapts decreases, which can be inferred from the adaptive algorithm. It is possible to identify four different phases in time step adaption for both CG and HDG discretizations. First, a constant time step phase used to kill the initial oscillations in the solution. The second phase lasts from 20 sec to around 40 sec, where the steady state solution is obtained with development of recirculation zones behind the wake of the cylinder. As the numerical round-offs and discretization errors are large enough to destabilize the flow, vortices are started to shred from the cylinder which characterises the third phase from 40 - 100 sec. Finally, once the vortex shedding is fully developed, the time step remains constant throughout the flow time. The value of time step in the last phase of the flow is provided in table 2.9. It can be noticed from the table 2.9 that the value of time step in final 40 sec for both HDG and CG are very close for the same level of local error tolerance. Finally, table 2.10 gives the lift and drag coefficients along with Strouhal number of the adaptive time stepping scheme analysis for both CG and HDG methods. It is

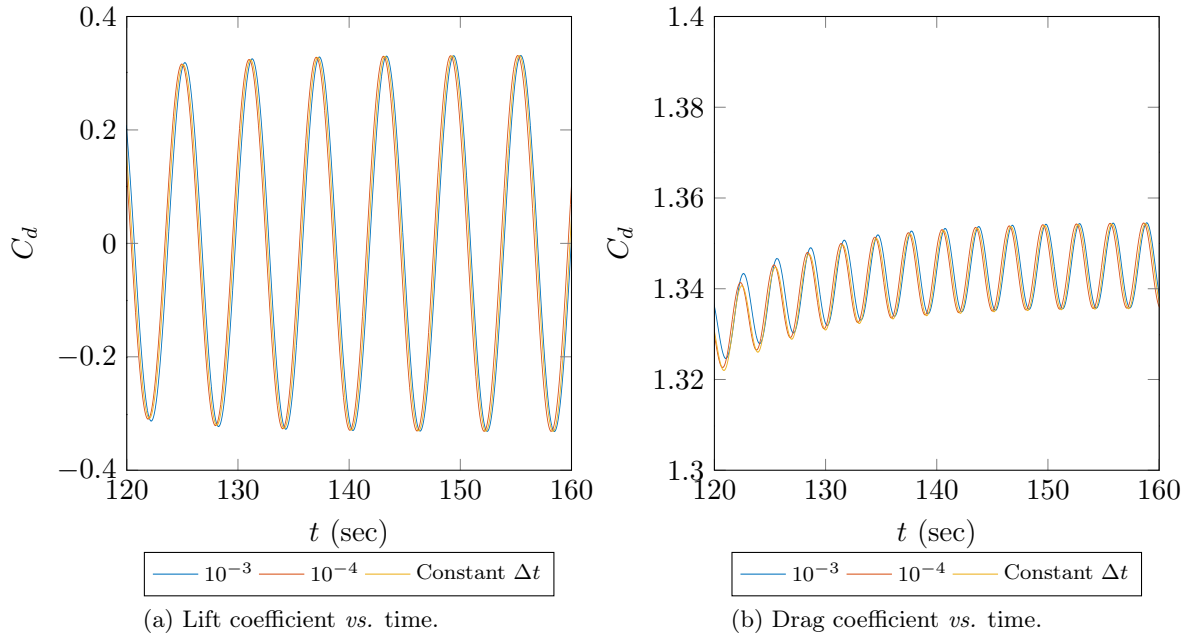


Figure 2.34: Flow around cylinder: variation of lift and drag coefficients with time for adaptive time stepping scheme for different levels of local error for HDG.

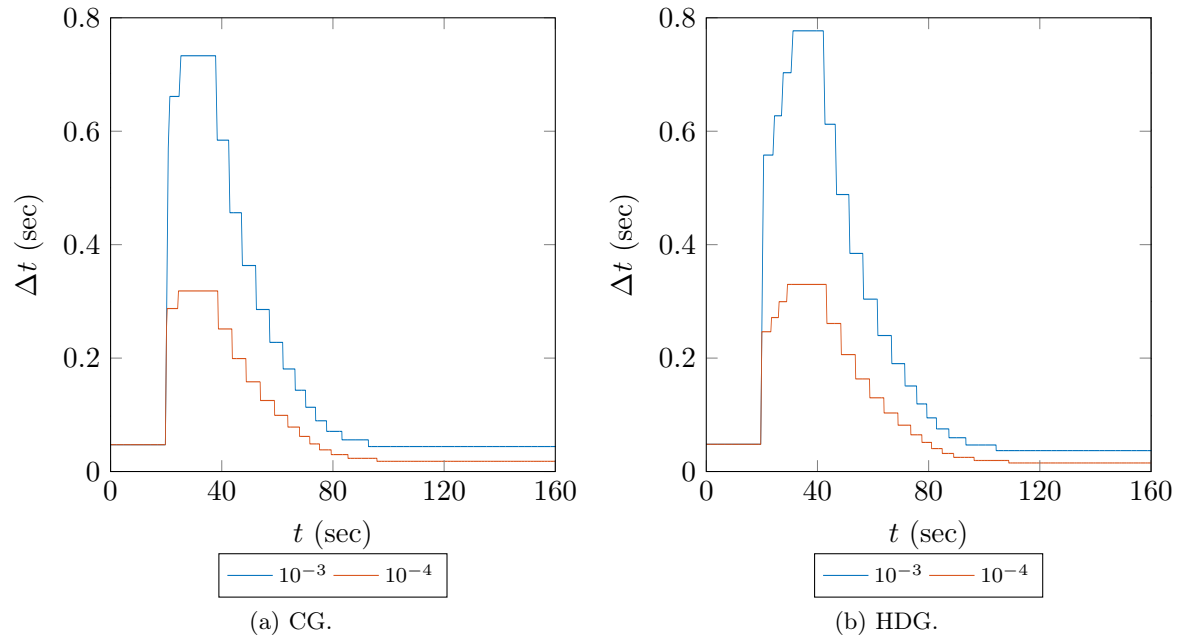


Figure 2.35: Flow around cylinder: variation of time step with flow time for adaptive time stepping scheme for different levels of local error for CG and HDG discretizations.

Table 2.9: Time step values in shedding regime for CG and HDG with adaptive time scheme.

Method	$\delta_{tol} = 10^{-3}$	$\delta_{tol} = 10^{-4}$
CG	0.0468	0.0209
HDG	0.0389	0.0173

Table 2.10: Lift, drag coefficients and Strouhal number for $Re = 100$ with adaptive time stepping scheme.

Variable	CG		HDG	
	$\delta_{tol} = 10^{-3}$	$\delta_{tol} = 10^{-4}$	$\delta_{tol} = 10^{-3}$	$\delta_{tol} = 10^{-4}$
C_l	0 ± 0.329	0 ± 0.329	0 ± 0.329	0 ± 0.329
C_d	1.344 ± 0.011	1.343 ± 0.011	1.339 ± 0.014	1.338 ± 0.015
St	0.164	0.164	0.164	0.164

evident from the table 2.10 that the values of lift and drag coefficients are very close to the ones obtained using constant time step analysis and also, literature data. Hence, it can be concluded that the proposed adaptive time stepping scheme is verified.

2.9 Conclusions

Theoretical DOF count is compared between HDG, with degree k for all variables, and CG with degree k for the velocity and $k - 1$ for the pressure. In the case of 2D, HDG has fewer DOFs than CG, when degree of approximation is more than 5, for both triangular and quadrilateral elements. In the case of 3D, they tend to be same only for very high degrees. However, when HDG with degree $k - 1$ is compared to CG with degree k for velocity and $k - 1$ for pressure, *i.e.*, with the same theoretical rates of convergence, the ratio of number of DOFs favours HDG when $k > 4$ for 2D and, in the case of 3D, $k > 6$ and $k > 12$ for hexahedral and tetrahedral elements, respectively. The number of non-zero entries in the global system and its factor are also provided for HDG and CG Stokes problems. HDG systems leads to fewer non zero entries when the degree of approximation, $k \geq 5$.

Then, Kovasznay flow, a benchmark for Stokes and Navier–Stokes problems, is considered to assess the computational efficiency between HDG and CG in 2D. It is noticed that, using a high-degree coarser mesh is computationally more efficient than using a low-degree finer mesh, with respect to CPU time for linear solver in the presence of smooth solutions. Numerical tests also show that HDG takes lesser CPU time for linear solver when compared to CG, for the same level of accuracy and for degree greater than 2.

The same comparison is carried out with NACA0012 airfoil example, with the error measured in the lift coefficient value. Again, HDG is more computationally efficient than CG for a given level of accuracy.

The condition numbers for HDG and CG are presented for Kovasznay flow and NACA airfoil examples. In most of the cases, HDG produces lower condition number values than CG.

A comparison of stability between HDG and CG is also presented, using a manufactured solution with a boundary layer. It is concluded that HDG can exhibit superior stability

properties than CG in the presence of sharp fronts, which occurs at high Reynolds numbers. It is also noticed that the increase of the parameter, τ_u , can enhance the stability of the HDG method, thus providing smooth converged solutions for Reynolds numbers much higher than CG.

The final part of the chapter deals with time discretization of unsteady Navier–Stokes equations using BDF schemes until order 3. The convergence rates in space and time are established for constant time step schemes using numerical experiments. The details of variable time stepping scheme are presented and verified using convergence tests. An adaptive time stepping scheme which estimates the local error with high-order BDF scheme to adapt the time step is presented. The example of flow around cylinder is considered to verify the adaptive time stepping scheme with both HDG and CG discretizations.

Chapter 3

Coupling of continuous and hybridizable discontinuous Galerkin methods

3.1 Overview

This chapter presents the coupled CG-HDG formulation for conjugate heat transfer problem. Firstly, the HDG formulation for Navier–Stokes presented in Chapter 2 is extended to coupled Navier–Stokes/convection-diffusion equations. Following, a coupled CG-HDG formulation is developed in the framework of heat conduction equation. The ideas from first two sections are combined to present the coupled CG-HDG formulation for conjugate heat transfer problem. Numerical results are presented in each section verifying the proposed formulations.

3.2 HDG formulation for coupled Navier–Stokes/convection-diffusion equations

3.2.1 Governing equations

In this section, HDG formulation introduced for incompressible Navier–Stokes equations in Chapter 2 is extended to coupled incompressible Navier–Stokes and convection-diffusion equations. Let Ω be the fluid domain with boundary $\partial\Omega$ divided into Dirichlet, $\partial\Omega_D$, and Neumann, $\partial\Omega_N$, boundaries. The steady state incompressible Navier–Stokes and convection-diffusion equations can be expressed as,

$$\begin{aligned} \operatorname{div}(\mathbf{u} \otimes \mathbf{u}) - \operatorname{div}(-p\mathbf{I} + \nu \operatorname{grad} \mathbf{u}) &= \mathbf{f}(\theta) + \bar{\mathbf{s}} && \text{in } \Omega, \\ \operatorname{div} \mathbf{u} &= 0 && \text{in } \Omega, \\ \operatorname{div}(-\alpha \operatorname{grad} \theta + \mathbf{u} \theta) &= \bar{g} && \text{in } \Omega, \\ \mathbf{u} &= \bar{\mathbf{u}} && \text{on } \partial\Omega_D, \\ \theta &= \bar{\theta} && \text{on } \partial\Omega_D, \\ (-p\mathbf{I} + \nu \operatorname{grad} \mathbf{u}) \mathbf{n} &= \bar{\mathbf{t}} && \text{on } \partial\Omega_N, \\ (-\alpha \operatorname{grad} \theta + \mathbf{u} \theta) \cdot \mathbf{n} &= \bar{q}_n && \text{on } \partial\Omega_N, \end{aligned} \tag{3.1}$$

where \mathbf{u} , θ and p are velocity, temperature and kinematic pressure, respectively. Material properties ν and α are kinematic viscosity and thermal diffusivity, respectively. $\mathbf{f}(\theta)$ is

the body force, which is a function of temperature, $\bar{\mathbf{s}}$ is an additional external body force acting on fluid and \bar{g} is the volumetric heat generation. $\bar{\mathbf{u}}$ and $\bar{\theta}$ are the prescribed velocity and temperature on the Dirichlet boundary, $\partial\Omega_D$, and $\bar{\mathbf{t}}$ and \bar{q}_n are the prescribed pseudo tractions and normal flux on the Neumann boundary, $\partial\Omega_N$.

As it is clear from the governing equations, this is a coupled system in \mathbf{u}, p and θ . The convective term $\mathbf{u} \cdot (\text{grad } \theta)$ in the convection-diffusion equation influences the temperature distribution and the body force $\mathbf{f}(\theta)$ in momentum equations governs the velocity of the fluid. Hence, a coupled system is established in both directions. In the present work, natural buoyancy flows are considered and Boussinesq approximation is used to compute the approximate body force, $\mathbf{f}(\theta)$, of Navier–Stokes equations. The artificial linear variation of density with temperature is expressed as,

$$\rho = \rho_0 (1 - \beta (\theta - \theta_0)), \quad (3.2)$$

where β is the thermal expansion coefficient and θ_0 is the reference temperature. The density is assumed to be constant to that of the reference state, ρ_0 . The gravitational force due to the artificial variation of the density can be expressed as the following body force vector per unit of mass of the reference state

$$\mathbf{f}(\theta) = \frac{\mathbf{g}(\rho - \rho_0)}{\rho_0} = -\mathbf{g}\beta(\theta - \theta_0), \quad (3.3)$$

where \mathbf{g} is the gravity acceleration vector. The important non-dimensional numbers that are used in the context of natural buoyancy flows are Rayleigh number (Ra) and Prandtl number (Pr), which are defined as,

$$Ra = \frac{g\beta L^3 \Delta\theta}{\nu\alpha}, \quad Pr = \frac{\nu}{\alpha}, \quad \text{where } \alpha = \frac{\kappa}{\rho_0 c_p}. \quad (3.4)$$

In equation (3.4), κ is heat conductivity, L and $\Delta\theta$ are the characteristic length and temperature difference in the domain, respectively.

3.2.2 Weak formulation

Using notation, $\mathbf{L} \approx \text{grad } \mathbf{u}$ and $\mathbf{q} \approx -\alpha \text{grad } \theta$, the HDG formulation of the coupled Navier–Stokes and convection-diffusion equations is: find $(\mathbf{u}, p, \mathbf{L}, \theta, \mathbf{q}, \hat{\mathbf{u}}, \rho_e, \hat{\theta}) \in [\mathcal{V}_k^h(\Omega)]^2 \times \mathcal{V}_k^h(\Omega) \times [\mathcal{V}_k^h(\Omega)]^{2 \times 2} \times \mathcal{V}_k^h(\Omega) \times [\mathcal{V}_k^h(\Omega)]^2 \times [\Lambda_k^h(\Gamma)]^2 \times \mathbb{R}^{n_{el}} \times \Lambda_k^h(\Gamma)$ satisfying the *local problem* in every element Ω^e ,

$$\begin{aligned} &(\delta \mathbf{L}, \mathbf{L})_{\Omega^e} + (\text{div } \delta \mathbf{L}, \mathbf{u})_{\Omega^e} - \langle \delta \mathbf{L} \mathbf{n}, \hat{\mathbf{u}} \rangle_{\partial\Omega^e} = 0, \\ &-(\text{grad } \delta \mathbf{u}, \mathbf{u} \otimes \mathbf{u})_{\Omega^e} - (\delta \mathbf{u}, \text{div } (-p\mathbf{I} + \nu \mathbf{L}))_{\Omega^e} \\ &\quad + \langle \delta \mathbf{u}, (\hat{\mathbf{u}} \otimes \hat{\mathbf{u}}) \mathbf{n} + \tau_u (\mathbf{u} - \hat{\mathbf{u}}) \rangle_{\partial\Omega^e} - (\delta \mathbf{u}, \mathbf{f}(\theta))_{\Omega^e} - (\delta \mathbf{u}, \bar{\mathbf{s}})_{\Omega^e} = 0, \end{aligned} \quad (3.5a)$$

$$\begin{aligned} &-(\text{grad } \delta p, \mathbf{u})_{\Omega^e} + \langle \delta p, \hat{\mathbf{u}} \cdot \mathbf{n} \rangle_{\partial\Omega^e} = 0, \\ &\frac{1}{|\partial\Omega^e|} \langle p, 1 \rangle_{\partial\Omega^e} = \rho_e, \end{aligned} \quad (3.5b)$$

$$\begin{aligned} &(\delta \mathbf{q}, \alpha^{-1} \mathbf{q})_{\Omega^e} - (\text{div } \delta \mathbf{q}, \theta)_{\Omega^e} + \langle \delta \mathbf{q} \cdot \mathbf{n}, \hat{\theta} \rangle_{\partial\Omega^e} = 0, \\ &(\delta \theta, \text{div } \mathbf{q})_{\Omega^e} - (\text{grad } \delta \theta, \mathbf{u} \theta)_{\Omega^e} + \langle \delta \theta, (\hat{\mathbf{u}} \cdot \mathbf{n} - \tau_\theta) \hat{\theta} \rangle_{\partial\Omega^e} \\ &\quad + \langle \delta \theta, \tau_\theta \theta \rangle_{\partial\Omega^e} - (\delta \theta, \bar{g})_{\Omega^e} = 0, \end{aligned} \quad (3.5c)$$

for $e = 1, \dots, n_{el}$, and the *global problem*

$$\sum_{e=1}^{n_{el}} \langle \delta \hat{\mathbf{u}}, (-p\mathbf{I} + \nu \mathbf{L}) \mathbf{n} + \tau_u (\hat{\mathbf{u}} - \mathbf{u}) \rangle_{\partial \Omega^e} = \langle \delta \hat{\mathbf{u}}, \bar{\mathbf{t}} \rangle_{\partial \Omega_N}, \quad (3.6a)$$

$$\langle \hat{\mathbf{u}} \cdot \mathbf{n}, 1 \rangle_{\partial \Omega^e} = 0 \quad \text{for } e = 1, \dots, n_{el},$$

$$\hat{\mathbf{u}} = \mathbb{P}_2(\bar{\mathbf{u}}) \quad \text{on } \partial \Omega_D,$$

$$\sum_{e=1}^{n_{el}} \langle \delta \hat{\theta}, (\mathbf{q} + \hat{\mathbf{u}} \hat{\theta}) \cdot \mathbf{n} + \tau_\theta (\theta - \hat{\theta}) \rangle_{\partial \Omega^e} = \langle \delta \hat{\theta}, \bar{q}_n \rangle_{\partial \Omega_N}, \quad (3.6b)$$

$$\hat{\theta} = \mathbb{P}_2(\bar{\theta}) \quad \text{on } \partial \Omega_D,$$

for all $(\delta \mathbf{u}, \delta p, \delta \mathbf{L}, \delta \theta, \delta \mathbf{q}, \delta \hat{\mathbf{u}}, \delta \hat{\theta}) \in [\mathcal{V}_k^h(\Omega)]^2 \times \mathcal{V}_k^h(\Omega) \times [\mathcal{V}_k^h(\Omega)]^{2 \times 2} \times \mathcal{V}_k^h(\Omega) \times [\mathcal{V}_k^h(\Omega)]^2 \times [\Lambda_k^h(\Gamma)]^2 \times \Lambda_k^h(\Gamma)$ such that $\delta \hat{\mathbf{u}} = \mathbf{0}$ and $\delta \hat{\theta} = 0$ on $\partial \Omega_D$, where the discrete spaces are defined in (1.2). $\mathbb{P}_2(\bar{\mathbf{u}})$ and $\mathbb{P}_2(\bar{\theta})$ are the \mathcal{L}_2 projections of the Dirichlet data into the approximation space on $\partial \Omega_D$. The parameters τ_u and τ_θ are positive and, following Nguyen *et al.* (2011c, 2009b), they are usually taken as

$$\tau_u \approx \frac{\nu}{L} + |\mathbf{u}|, \quad \tau_\theta \approx \frac{\alpha}{L} + |\mathbf{u} \cdot \mathbf{n}|, \quad (3.7)$$

where L is the characteristic length of the problem. Even though the so-called stabilization parameter has some influence on the accuracy of the HDG solution, the method is very robust in front of variations of τ_u and τ_θ (Nguyen *et al.*, 2009b,a). As stated in Chapter 2, these parameters can have an important effect on stability properties of the method. The numerical results in Section 3.2.4 discusses the influence of these parameters on the convergence of the solution.

The discretization of local and global problems (3.5)-(3.6) leads to a discrete residual presented in (3.10). This coupled system is solved in a monolithic sense using Newton–Raphson method computing the exact Jacobian matrix (3.11). As already explained in previous sections, static condensation is used to express the local variables in terms of global variables and solve the final system in only global unknowns. The constraint in (3.5b) is applied using a Lagrangian multiplier λ_e on each element. These are gathered in the vector $\boldsymbol{\lambda}$. The definition of elemental matrices and tangent operators are presented in Appendix C.

The HDG formulation provides a numerical solution with optimal convergence of order $k+1$ (k is the degree of approximation) in \mathcal{L}_2 norm for velocity, \mathbf{u} , pressure, p , gradient of velocity, \mathbf{L} , temperature, θ , and flux, \mathbf{q} . As the mean of the velocity and mean of the temperature inside each element converges with order $k+2$, new approximate solutions, \mathbf{u}^* and θ^* , can be computed element-by-element which converge with order $k+2$. The details of the post-processing for Navier–Stokes is presented in Chapter 2. The post-processed solution of the convection-diffusion equation can be obtained by solving following equations,

$$\begin{aligned} -\operatorname{div}(\alpha \operatorname{grad} \theta^*) &= \operatorname{div} \mathbf{q} && \text{in } \Omega^e, \\ -(\alpha \operatorname{grad} \theta^*) \cdot \mathbf{n} &= \mathbf{q} \cdot \mathbf{n} && \text{on } \partial \Omega^e, \\ (\theta^*, 1)_{\Omega^e} &= (\theta, 1)_{\Omega^e}. \end{aligned} \quad (3.8)$$

This induces a weak problem in a richer finite dimensional space, *i.e.*, find $\theta^* \in [\mathcal{V}_{k+1}^h(\Omega)]$ such that

$$\begin{aligned} -(\operatorname{grad} \delta \theta^*, \alpha \operatorname{grad} \theta^*)_{\Omega^e} &= (\operatorname{grad} \delta \theta^*, \mathbf{q})_{\Omega^e}, \\ (\theta^*, 1)_{\Omega^e} &= (\theta, 1)_{\Omega^e}, \end{aligned} \quad (3.9)$$

$$\begin{bmatrix} \begin{Bmatrix} \mathbf{r}_{\hat{u}} \\ \mathbf{r}_{\rho} \\ \mathbf{r}_{\hat{\theta}} \end{Bmatrix} \\ \begin{Bmatrix} \mathbf{r}_u \\ \mathbf{r}_L \\ \mathbf{r}_p \\ \mathbf{r}_{\lambda} \\ \mathbf{r}_{\theta} \\ \mathbf{r}_q \end{Bmatrix} \end{bmatrix} = \begin{bmatrix} \begin{matrix} \mathbf{A}_{\hat{u}\hat{u}} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{\rho\hat{u}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{\hat{\theta}\hat{\theta}} + \mathbf{C}_{\hat{\theta}\hat{\theta}}(\hat{\mathbf{u}}) \end{matrix} & \begin{matrix} \mathbf{A}_{\hat{u}u} & \mathbf{A}_{\hat{u}L} & \mathbf{A}_{\hat{u}p} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{matrix} \\ \hline \begin{matrix} \mathbf{A}_{u\hat{u}} + \mathbf{C}_{u\hat{u}}(\hat{\mathbf{u}}) & \mathbf{0} \\ \mathbf{A}_{L\hat{u}} & \mathbf{0} \\ \mathbf{A}_{p\hat{u}} & \mathbf{0} \\ \mathbf{0} & -\mathbf{1} \\ \mathbf{0} & \mathbf{A}_{\theta\hat{\theta}} + \mathbf{C}_{\theta\hat{\theta}}(\hat{\mathbf{u}}) \\ \mathbf{0} & \mathbf{A}_{q\hat{\theta}} \end{matrix} & \begin{matrix} \mathbf{A}_{uu} + \mathbf{C}_{uu}(\mathbf{u}) & \mathbf{A}_{uL} & \mathbf{A}_{up} \\ \mathbf{A}_{Lu} & \mathbf{A}_{LL} & \mathbf{0} \\ \mathbf{A}_{pu} & \mathbf{0} & \mathbf{A}_{pp}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{pp} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{matrix} \end{bmatrix} \begin{bmatrix} \begin{Bmatrix} \hat{\mathbf{u}} \\ \rho \\ \hat{\theta} \end{Bmatrix} \\ \begin{Bmatrix} \mathbf{u} \\ \mathbf{L} \\ \mathbf{p} \\ \lambda \\ \theta \\ \mathbf{q} \end{Bmatrix} \end{bmatrix} - \begin{bmatrix} \begin{Bmatrix} \bar{\mathbf{t}} \\ \mathbf{0} \\ \bar{\mathbf{q}}_n \end{Bmatrix} \\ \begin{Bmatrix} \mathbf{f}_{\theta_0} + \bar{\mathbf{s}} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \bar{\mathbf{g}} \\ \mathbf{0} \end{Bmatrix} \end{bmatrix} = \mathbf{0}, \quad (3.10)$$

$$\begin{bmatrix} \begin{matrix} \mathbf{A}_{\hat{u}\hat{u}} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{\rho\hat{u}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{\hat{\theta}\hat{\theta}} + \mathbf{C}_{\hat{\theta}\hat{\theta}}(\hat{\mathbf{u}}) \end{matrix} & \begin{matrix} \mathbf{A}_{\hat{u}u} & \mathbf{A}_{\hat{u}L} & \mathbf{A}_{\hat{u}p} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{matrix} \\ \hline \begin{matrix} \mathbf{A}_{u\hat{u}} + \mathbf{C}_{u\hat{u}}(\hat{\mathbf{u}}) & \mathbf{0} \\ \mathbf{A}_{L\hat{u}} & \mathbf{0} \\ \mathbf{A}_{p\hat{u}} & \mathbf{0} \\ \mathbf{0} & -\mathbf{1} \\ \mathbf{0} & \mathbf{A}_{\theta\hat{\theta}} + \mathbf{C}_{\theta\hat{\theta}}(\hat{\mathbf{u}}) \\ \mathbf{0} & \mathbf{A}_{q\hat{\theta}} \end{matrix} & \begin{matrix} \mathbf{A}_{uu} + \mathbf{C}_{uu}(\mathbf{u}) & \mathbf{A}_{uL} & \mathbf{A}_{up} \\ \mathbf{A}_{Lu} & \mathbf{A}_{LL} & \mathbf{0} \\ \mathbf{A}_{pu} & \mathbf{0} & \mathbf{A}_{pp}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{pp} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{matrix} \end{bmatrix} \begin{bmatrix} \begin{Bmatrix} \delta\hat{\mathbf{u}} \\ \delta\rho \\ \delta\hat{\theta} \end{Bmatrix} \\ \begin{Bmatrix} \delta\mathbf{u} \\ \delta\mathbf{L} \\ \delta\mathbf{p} \\ \delta\lambda \\ \delta\theta \\ \delta\mathbf{q} \end{Bmatrix} \end{bmatrix} = - \begin{bmatrix} \begin{Bmatrix} \mathbf{r}_{\hat{u}} \\ \mathbf{r}_{\rho} \\ \mathbf{r}_{\hat{\theta}} \end{Bmatrix} \\ \begin{Bmatrix} \mathbf{r}_u \\ \mathbf{r}_L \\ \mathbf{r}_p \\ \mathbf{r}_{\lambda} \\ \mathbf{r}_{\theta} \\ \mathbf{r}_q \end{Bmatrix} \end{bmatrix}. \quad (3.11)$$

for all $\delta\theta^* \in [\mathcal{V}_{k+1}^h(\Omega)]$ and $i = 1, \dots, n_{el}$, where $\mathcal{V}_{k+1}^h(\Omega)$ must be a bigger space with one degree more than in $\mathcal{V}_k^h(\Omega)$. θ^* converges asymptotically with a rate of $k + 2$ in the \mathcal{L}_2 norm for a mesh with uniform degree, k .

Numerical experiments in the following section show that the property of super-convergence is retained in the case of the Navier–Stokes/convection-diffusion coupled formulation for both velocity and temperature.

3.2.3 Convergence results

In this section, the convergence results of Navier–Stokes equations coupled with convection-diffusion equation are presented for HDG. The solution of Navier–Stokes equations is taken as Kovasznay flow (Kovasznay, 1948) in the domain $[0, 2] \times [-0.5, 1.5]$ presented in equation (2.20). The analytical solution considered for the temperature variation is of sinusoidal form and it is given as follows,

$$\theta = \sin\left(\frac{3\pi x_1}{4}\right) \sin\left(\frac{3\pi x_2}{4} + \frac{3\pi}{8}\right). \quad (3.12)$$

Dirichlet boundary conditions are prescribed for velocity and temperature on all the exterior boundary. The solution is computed at $Re = 20$ and the thermal diffusivity, α , is taken as unity.

As discussed earlier, the Boussinesq approximation term couples the Navier–Stokes and convection-diffusion equations. In the Boussinesq term (3.3), β is taken as unity, $\mathbf{g} = -10\mathbf{e}_2$ and $\theta_0 = 0$. An appropriate body force term, $\bar{\mathbf{s}}$ in momentum equation, and heat generation, \bar{g} , in the convection-diffusion equation are computed from the analytical solution. The body force term, $\bar{\mathbf{s}}$, has the non-zero contribution only in the convergence analysis. The stabilization parameters τ_u and τ_θ are both taken as unity on all faces of each element in this example.

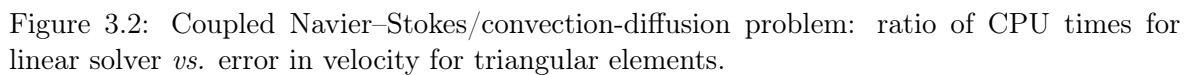
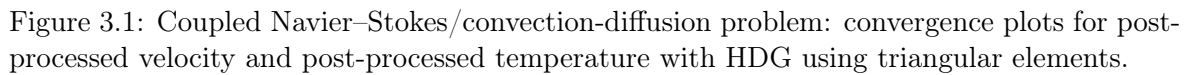
Figure 3.1 shows the convergence of post-processed solution of velocity and temperature for triangular meshes for degree $k = 2 - 9$ and element size, $h = 2/\{2, 2^2, 2^3, 2^4, 2^5, 2^6\}$. It is clear that the super-convergence of HDG is retained for both velocity and temperature in the coupled framework.

As the extension of the work in Chapter 2, the results of computational efficiency of HDG (T_k) and CG ($T_k T_{k-1}$) are presented for the coupled Navier–Stokes/convection-diffusion equations. Here, only the plots of the ratio of CPU times of HDG to CG against the error are presented. Figures 3.2 and 3.3 shows the ratio of CPU times with respect to error in velocity and ratio of CPU times with respect to error in temperature, respectively. As explained in the Section 2.6, the errors values from CG are used in plots and CPU times of HDG are interpolated corresponding to the errors of CG. Note that post-processed solutions are used to compute errors in HDG for both velocity and temperature. From the results, it can be concluded that HDG outperforms CG in terms of CPU time for linear solver for a given level accuracy and $k \geq 3$. Hence, the conclusions drawn from the computational efficiency for Navier–Stokes can be extended to the present coupled Navier–Stokes/convection-diffusion analysis.

3.2.4 Rayleigh–Bénard convection flow

This section presents the results of Rayleigh–Bénard convection flow at different Rayleigh numbers. This is a standard benchmark example for natural buoyancy flows.

A square cavity of unit length is considered with a temperature gradient along x_1 –direction. The top and bottom walls are prescribed with adiabatic boundary conditions, whereas the



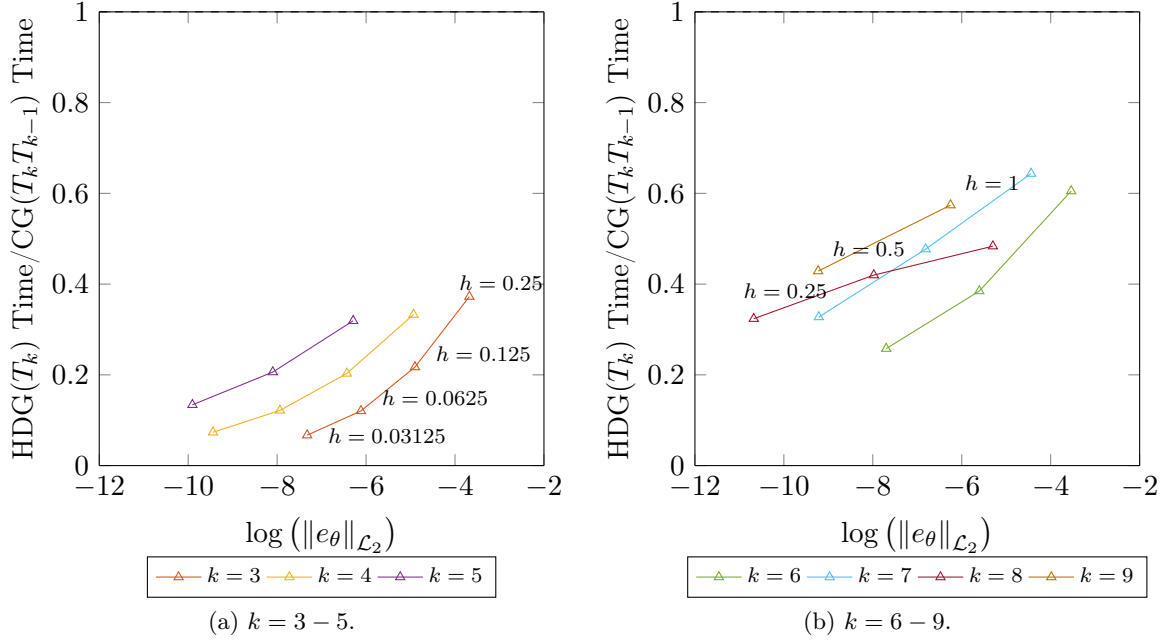


Figure 3.3: Coupled Navier–Stokes/convection-diffusion problem: ratio of CPU times for linear solver *vs.* error in temperature for triangular elements.

Table 3.1: Rayleigh–Bénard convection flow: stabilization parameters used in the analysis.

Ra	$10^3, 10^4$	10^5	10^6	$10^7, 10^8$
τ_u, τ_θ	1	10	100	200

lateral walls are prescribed with Dirichlet boundary conditions for temperature. No-slip conditions are applied on all the boundary for the velocity. Rayleigh numbers ranging from 10^3 to 10^8 , along with Prandtl number (Pr) of 0.71, are considered in the analysis. The temperature difference is kept at unity with left wall at 0.5 and right wall at -0.5 , $\mathbf{g} = -10\mathbf{e}_2$ and β is chosen based on the Ra . Nusselt number at the hot wall is used to compare the HDG (T_k) solution with CG ($T_k T_{k-1}$) and other results from literature. Average Nusselt number is defined as,

$$\overline{Nu} = -\frac{1}{\theta_h - \theta_c} \int_0^L \frac{\partial \theta}{\partial x_1} ds, \quad (3.13)$$

where θ_h, θ_c denotes the temperatures on the hot and cold walls, respectively and L is the length of the cavity.

A uniform mesh with triangular elements of degree $k = 5$ and element size $h = 1/2^6$ is chosen. All the results presented corresponds to steady state analysis. Sometimes, at higher Rayleigh number, convergence to the solution could not be achieved in a single step. In this case, Ra number is increased in a finite number of steps to obtain a converged solution. No stabilization techniques for the CG method are used.

The stabilization parameters in the case of HDG discretization are critical in obtaining a solution at higher Ra . Table 3.1 provides the values of parameters used in the present work for different Ra . The analytical maximum velocity in the domain increases with increasing

Table 3.2: Rayleigh–Bénard convection flow: summary of important quantities and comparison with literature values for $Ra = 10^3 - 10^5$. The velocities of HDG correspond to post-processed solutions.

Ra	Quantity	De Vahl Davis (1983)	HDG (T_5) / CG (T_5T_4)
10^3	\overline{Nu}	1.117	1.117
	Max. u_1 along $x_1 = 0.5$	3.649	3.649
	Max. u_2 along $x_2 = 0.5$	3.697	3.697
10^4	\overline{Nu}	2.238	2.244
	Max. u_1 along $x_1 = 0.5$	16.178	16.183
	Max. u_2 along $x_2 = 0.5$	19.617	16.626
10^5	\overline{Nu}	4.509	4.521
	Max. u_1 along $x_1 = 0.5$	34.73	34.740
	Max. u_2 along $x_2 = 0.5$	68.59	68.632

Ra number. In the present work, the stabilization parameters are chosen to be in the order of magnitude of the maximum velocity in the domain for a given Ra .

Figure 3.4 presents the isolines of velocity and temperature at Ra of $10^4, 10^6$ and 10^8 . It can be noticed from the plots that as the Ra number increases, a strong boundary layer is formed at both lateral edges of the domain. The vertical isotherms denote that the dominant mechanism of heat transfer is conduction and as the isotherms depart from the vertical position, convection becomes the dominant form of heat transfer. It is evident from the velocity isolines that the velocity magnitude varies from 20 at $Ra = 10^4$ to around 2200 at $Ra = 10^8$.

The variation of dimensionless u_1 along $x_1 = 0.5$ and dimensionless u_2 along $x_2 = 0.5$ are presented in figs. 3.5a and 3.5b, respectively, to show the evolution of boundary layer as the Ra increases. Similarly, the variation of temperature along $x_2 = 0.5$ is plotted for all the Ra numbers considered in fig. 3.5c. The present numerical results are compared with Betts and Haroutunian (1983) whenever applicable and the values are denoted by circle marks in the plots. The fact that HDG could resolve the boundary layer without any refinement on boundary shows the effectiveness of using high-order elements.

To validate the results obtained in the present work, various quantities of interest are compared with the results from literature. The comparison is presented in the tables 3.2 and 3.3, which shows the results of HDG, CG and the literature ones. The table presents the results of both HDG and CG with that of literature ones. The results from HDG and CG are practically similar owing to the high-degree approximations and relatively fine mesh. Nevertheless, HDG produces slightly more accurate solution than CG when Nusselt number is compared for $Ra = 10^8$. In addition, when CPU times for the linear solver are considered, HDG outperforms CG in the present example using a uniform triangular mesh with $h = 1/2^6$ and $k = 5$. Table 3.4 gives the details of computations for HDG and CG at $Ra = 10^5$. Even though, HDG has slightly more DOFs than CG, the CPU time for the linear solver per one non-linear iteration of HDG is less than its CG counterpart. This can be attributed towards the regular block structure of the HDG stiffness matrix in accordance with the study in Kirby *et al.* (2012), Yakovlev *et al.* (2016). This fact is more pronounced by comparing the condition numbers of the matrix, which is one order of magnitude larger for CG than for HDG, for same mesh and degree of approximation.

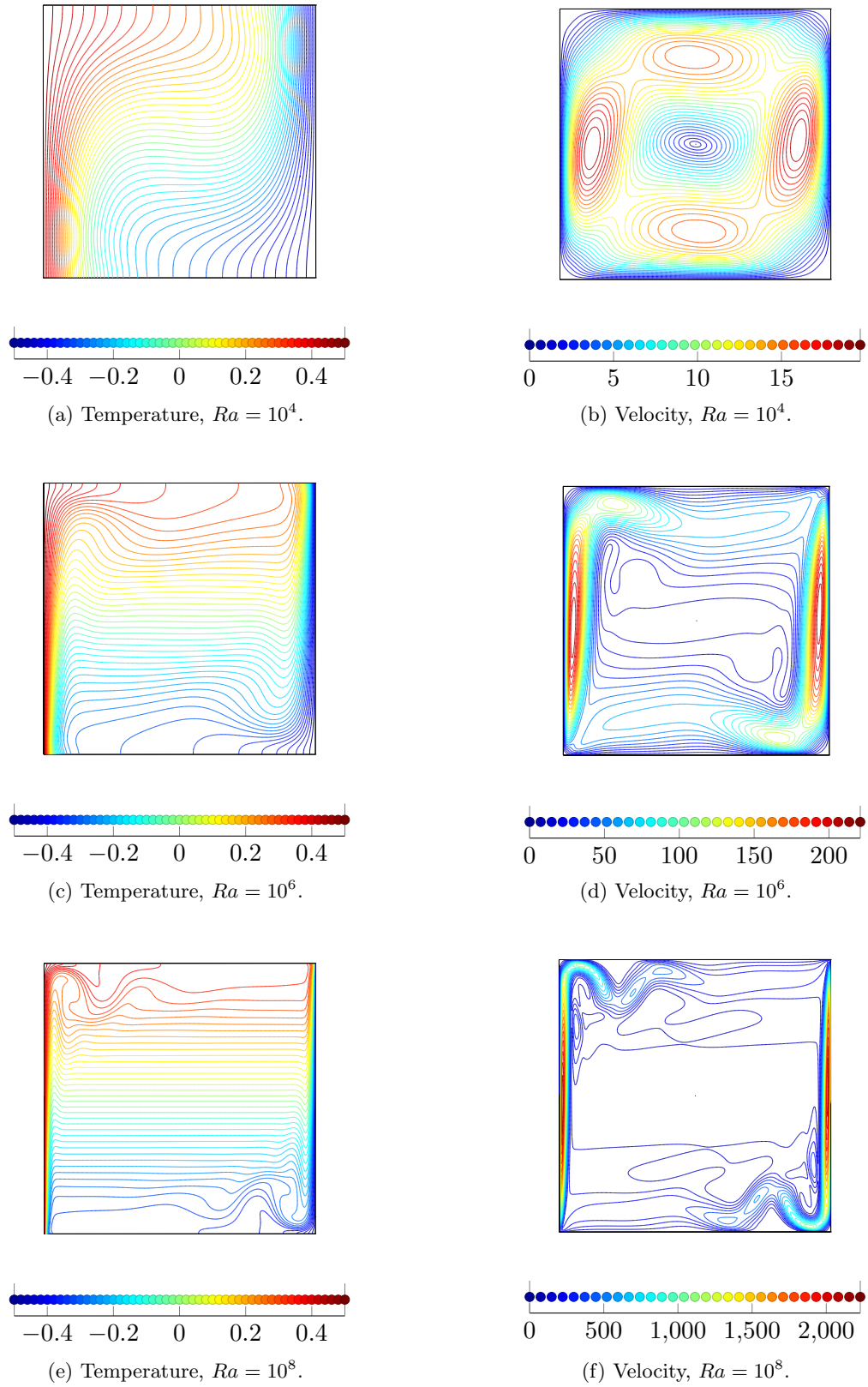


Figure 3.4: Rayleigh–Bénard convection flow: isolines of post-processed solutions of temperature and velocity at different Ra numbers using HDG.

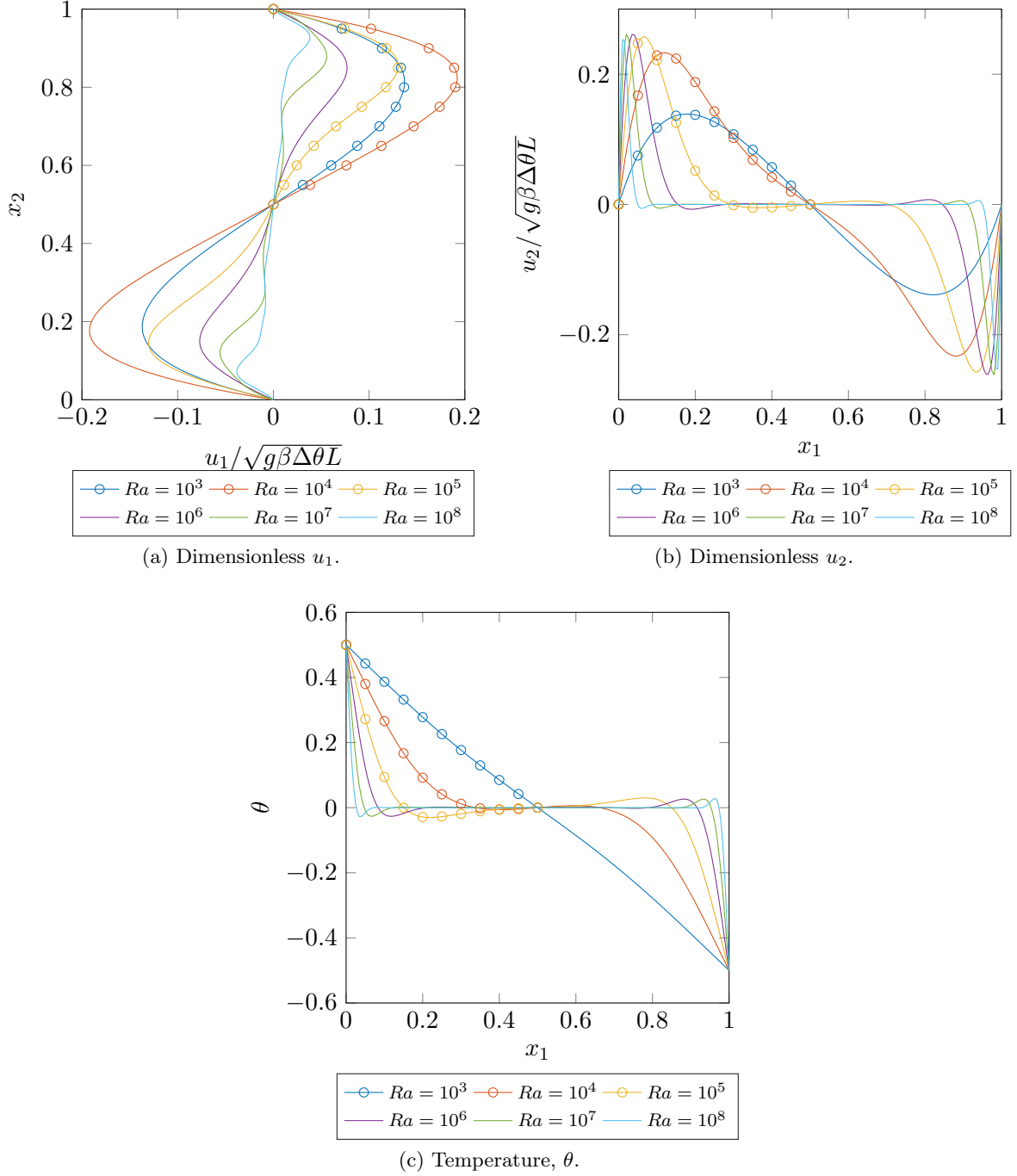


Figure 3.5: Rayleigh-Bénard convection flow: distribution of post-processed temperature and velocity components at different Ra numbers using HDG. Circles correspond to reference values in Betts and Haroutunian (1983).

Table 3.3: Rayleigh–Bénard convection flow: summary of important quantities and comparison with literature values for $Ra = 10^6 - 10^8$. The velocities of HDG correspond to post-processed solutions.

Ra	Quantity	Le Quéré (1991)	HDG (T_5)	CG (T_5T_4)
10^6	\overline{Nu}	8.825	8.825	8.825
	Max. u_1 along $x_1 = 0.5$	64.83	64.826	64.826
	Max. u_2 along $x_2 = 0.5$	220.6	220.390	220.390
10^7	\overline{Nu}	16.523	16.523	16.521
	Max. u_1 along $x_1 = 0.5$	148.580	148.583	148.583
	Max. u_2 along $x_2 = 0.5$	699.236	695.940	695.940
10^8	\overline{Nu}	30.225	30.209	30.145
	Max. u_1 along $x_1 = 0.5$	321.876	321.567	321.568
	Max. u_2 along $x_2 = 0.5$	2222.39	2221.647	2221.657

Table 3.4: Rayleigh–Bénard convection flow: computational details for HDG and CG for $Ra = 10^5$ and $h = 1/2^6$.

	HDG (T_5)	CG (T_5T_4)
No. of DOFs, ndof	2 31 680	2 03 140
CPU time for linear solver	67.8 sec	75.0 sec
No. of non-linear iterations	13	13
Condition number, $\kappa(\mathbf{A})$	$\mathcal{O}(10^{11})$	$\mathcal{O}(10^{12})$
No. of non-zeros, nnz	20 478 391	19 854 033

Hence, it can be advantageous to use high-order HDG methods to solve the coupled Navier–Stokes/convection-diffusion equations, especially in the presence of strong boundary layers.

3.3 CG-HDG coupled formulation for the heat equation

3.3.1 Domain representation

Let the domain, Ω , be split into two sub-domains, Ω_1 and Ω_2 , such that $\bar{\Omega} = \bar{\Omega}_1 \cup \bar{\Omega}_2$ with an interface $\Gamma_I = \bar{\Omega}_1 \cap \bar{\Omega}_2$, as shown in fig. 3.6. In physical terms, Ω_1 corresponds to fluid domain and Ω_2 represents the solid part in a conjugate heat transfer problem framework. In this work, either HDG or CG discretization can be considered in Ω_1 and Ω_2 . When HDG is considered in Ω_1 and CG in Ω_2 , the coupled model will be referred as CG-HDG model, the model will be denoted by CG-CG model when CG discretization is considered in Ω_1 and Ω_2 and HDG-HDG model uses HDG discretization in both Ω_1 and Ω_2 .

The domain Ω is assumed to be divided into n_{el} elements, Ω^e , with the boundaries $\partial\Omega^e$.

$$\bar{\Omega} = \bigcup_{e=1}^{n_{el}} \bar{\Omega}^e, \quad \Omega^e \cap \Omega^k = \emptyset \quad \text{for } e \neq k, \quad (3.14)$$

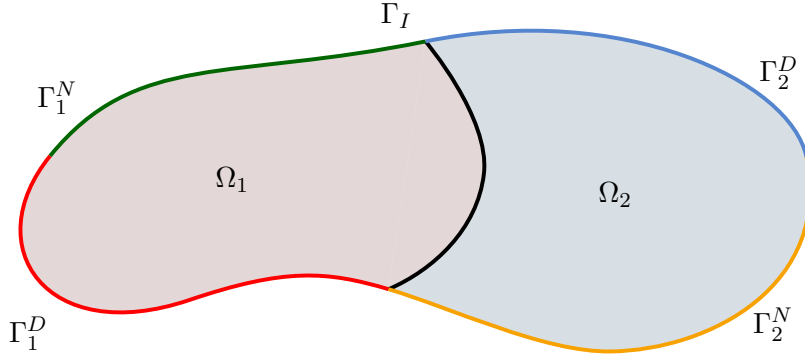


Figure 3.6: Domain representation: Domain discretization used to present coupled CG-HDG formulation.

The elements in Ω_1 and Ω_2 are represented as Ω_1^e and Ω_2^e , respectively, while the exterior boundaries are denoted by $\Gamma_1 = \Gamma_1^D \cup \Gamma_1^N$ and $\Gamma_2 = \Gamma_2^D \cup \Gamma_2^N$, respectively. The superscripts D and N in the notation of exterior boundaries stands for Dirichlet and Neumann parts, respectively. The union of all edges in Ω_1 is denoted as

$$\Gamma_1 = \bigcup_{e=1}^{m_{el}} \partial\Omega_1^e, \quad (3.15)$$

where m_{el} is the number of elements in Ω_1 . Similarly, the union of all edges in Ω_2 is denoted as

$$\Gamma_2 = \bigcup_{e=1}^{p_{el}} \partial\Omega_2^e, \quad (3.16)$$

where p_{el} is the number of elements in Ω_2 . Hence, total number of elements in Ω is $n_{el} = m_{el} + p_{el}$.

3.3.2 Governing equations

The heat equation in Ω_1 and Ω_2 , along with the transmission conditions on Γ_I , are

$$\mathbf{q}_1 + (\alpha_1 \text{grad } \theta_1) = \mathbf{0} \quad \text{in } \Omega_1, \quad (3.17a)$$

$$\text{div } \mathbf{q}_1 = \bar{g}_1 \quad \text{in } \Omega_1, \quad (3.17b)$$

$$-\text{div } (\alpha_2 \text{grad } \theta_2) = \bar{g}_2 \quad \text{in } \Omega_2, \quad (3.17c)$$

$$\theta_1 = \bar{\theta}_1 \quad \text{on } \Gamma_1^D, \quad (3.17d)$$

$$\theta_2 = \bar{\theta}_2 \quad \text{on } \Gamma_2^D, \quad (3.17e)$$

$$\theta_1 - \theta_2 = 0 \quad \text{on } \Gamma_I, \quad (3.17f)$$

$$\mathbf{q}_1 \cdot \mathbf{n}_1 - (\alpha_2 \text{grad } \theta_2) \cdot \mathbf{n}_2 = 0 \quad \text{on } \Gamma_I, \quad (3.17g)$$

where θ_1 and θ_2 are the temperatures in Ω_1 and Ω_2 , respectively and \mathbf{q}_1 is the independently approximated flux in Ω_1 . Thermal diffusivity coefficients are denoted by α_1 , α_2 , heat generations per unit volume are given by \bar{g}_1 , \bar{g}_2 , where the subscripts 1 and 2 denote that quantities are defined in Ω_1 and Ω_2 , respectively. Unit normal vectors on Γ_I , \mathbf{n}_1 and \mathbf{n}_2 , are outward vectors to Ω_1 and Ω_2 , respectively, which satisfy $\mathbf{n}_1 = -\mathbf{n}_2$. Dirichlet boundary conditions are prescribed for both sub-domains on the exterior boundaries to simplify the presentation,

i.e., $\Gamma_1^N = \emptyset$ and $\Gamma_2^N = \emptyset$. The extension of the formulation to problems including Neumann boundary conditions on the exterior boundary is straightforward following the usual procedure for HDG or CG formulations. Equation (3.17f) represents the continuity of θ , whereas (3.17g) states the equilibrium of the normal flux across the interface.

3.3.3 Strong formulation of the CG-HDG coupled problem

The strong form of the governing equations in Ω_2 can be written as,

$$\begin{aligned} -\operatorname{div}(\alpha_2 \operatorname{grad} \theta_2) &= \bar{g}_2 & \text{in } \Omega_2, \\ \theta_2 &= \bar{\theta}_2 & \text{on } \Gamma_2^D. \end{aligned} \quad (3.18)$$

As discussed in Chapter 2, the strong form for any HDG discretization involves two sub-problems namely, local and global problems. The local problem can be stated inside each element using trace variable, $\hat{\theta}_1$, as the Dirichlet boundary condition. Hence, strong form of the *local* problem in an element, Ω_1^e , can be expressed as,

$$\begin{aligned} \mathbf{q}_1 + (\alpha_1 \operatorname{grad} \theta_1) &= \mathbf{0} & \text{in } \Omega_1^e, \\ \operatorname{div} \mathbf{q}_1 &= \bar{g}_1 & \text{in } \Omega_1^e, \\ \theta_1 &= \hat{\theta}_1 & \text{on } \partial\Omega_1^e. \end{aligned} \quad (3.19)$$

The *global* problem in HDG states the so-called conservativity condition, *i.e.*, continuity of the normal fluxes along the interior edges of the mesh, Γ_1 . It can be stated as follows,

$$\begin{aligned} \llbracket \mathbf{q}_1 \cdot \mathbf{n} \rrbracket &= 0 & \text{on } \Gamma_1, \\ \hat{\theta}_1 &= \bar{\theta}_1 & \text{on } \Gamma_1^D. \end{aligned} \quad (3.20)$$

As it can be noticed from (3.20), the Dirichlet boundary conditions are applied on the global problem in Ω_1 . The transmission conditions along the interface, Γ_I , can be stated as follows,

$$\begin{aligned} \hat{\theta}_1 &= \theta_2, \\ \mathbf{q}_1 \cdot \mathbf{n}_1 &= (\alpha_2 \operatorname{grad} \theta_2) \cdot \mathbf{n}_2, \end{aligned} \quad (3.21)$$

The following section presents the weak formulations of the presented equations.

3.3.4 Weak formulation of the CG-HDG coupled problem

As shown in fig. 3.7, in Ω_2 the temperature field, θ_2 , is approximated with a continuous space on the mesh represented in blue, while in the HDG domain, Ω_1 , the elemental variables θ_1 and \mathbf{q}_1 are approximated within each element represented in green and a new independently approximated trace variable, $\hat{\theta}_1$, is defined along the red edges (mesh skeleton).

The CG weak form of the heat equation in Ω_2 is,

$$(\operatorname{grad} \delta\theta_2, \alpha_2 \operatorname{grad} \theta_2)_{\Omega_2} - \langle \delta\theta_2, \alpha_2 (\operatorname{grad} \theta_2 \cdot \mathbf{n}_2) \rangle_{\Gamma_I} - (\delta\theta_2, \bar{g}_2)_{\Omega_2} = 0, \quad (3.22)$$

where $\delta\theta_2 = 0$ on Γ_2^D . Equation (3.22) is obtained after multiplying equation (3.17c) with $\delta\theta_2$ and performing integration by parts.

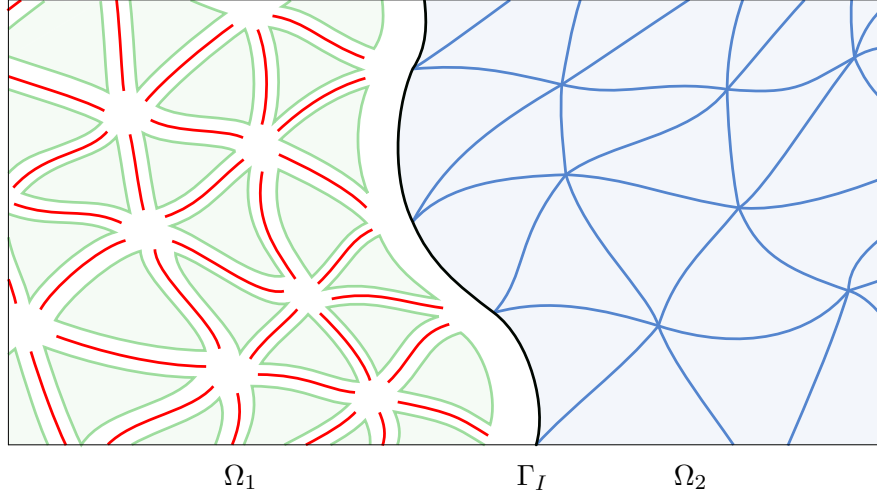


Figure 3.7: Representation of a computational mesh for coupled discretization. *Green* triangles represent the HDG local elemental variables while the *red* edges correspond to HDG trace variable. CG mesh is represented in *blue* and Γ_I is represented in *black*.

For the HDG domain, Ω_1 , the discrete problem is expressed as element-by-element local problems and the so-called global problem (Cockburn *et al.*, 2009b). The *local problem* in each element is,

$$\begin{aligned}
 (\delta\theta_1, \operatorname{div} \mathbf{q}_1)_{\Omega_1^e} + \langle \delta\theta_1, \tau(\theta_1 - \hat{\theta}_1) \rangle_{\partial\Omega_1^e \setminus \Gamma_I} \\
 + \langle \delta\theta_1, \tau(\theta_1 - \theta_2) \rangle_{\partial\Omega_1^e \cap \Gamma_I} - (\delta\theta_1, \bar{g}_1)_{\Omega_1^e} = 0, \\
 (\delta\mathbf{q}_1, \alpha_1^{-1} \mathbf{q}_1)_{\Omega_1^e} - (\operatorname{div} \delta\mathbf{q}_1, \theta_1)_{\Omega_1^e} + \langle \delta\mathbf{q}_1 \cdot \mathbf{n}, \hat{\theta}_1 \rangle_{\partial\Omega_1^e \setminus \Gamma_I} \\
 + \langle \delta\mathbf{q}_1 \cdot \mathbf{n}_1, \theta_2 \rangle_{\partial\Omega_1^e \cap \Gamma_I} = 0,
 \end{aligned} \tag{3.23}$$

where $\hat{\theta}_1$ is an independently approximated trace variable along the mesh skeleton, Γ_1 , which is represented in red in fig. 3.7, and τ is a stabilization parameter of order $\mathcal{O}(\alpha_1)$. Parameter τ has an important effect on stability, accuracy and convergence properties of the HDG method (Cockburn *et al.*, 2009a). As usual in HDG, the local problem can be solved element-by-element to express θ_1 and \mathbf{q}_1 in terms of $\hat{\theta}_1$ and, in the present case, θ_2 as well. Note that the only difference of equations (3.23) with the standard HDG local problem is that the Dirichlet data, that is imposed in weak is

$$\theta_1 = \begin{cases} \hat{\theta}_1 & \text{on } \partial\Omega_1^e \setminus \Gamma_I, \\ \theta_2 & \text{on } \partial\Omega_1^e \cap \Gamma_I. \end{cases} \tag{3.24}$$

For the elements along the interface, no trace variables are considered, as illustrated in fig. 3.7. For elements in the interior of Ω_1 , the local problem is the standard one, with a weak imposition of $\theta_1 = \hat{\theta}_1$ on $\partial\Omega_1^e$. The Dirichlet boundary condition (3.24) ensures the weak continuity of the temperature, *i.e.*, the transmission condition (3.17f) on Γ_I .

The *global problem* in Ω_1 is the usual HDG global problem, which can be presented as

$$\sum_{e=1}^{m_{el}} \langle \delta\hat{\theta}_1, \hat{\mathbf{q}}_1 \cdot \mathbf{n} \rangle_{\partial\Omega_1^e \setminus \Gamma_I} = 0, \tag{3.25a}$$

$$\hat{\theta}_1 = \mathbb{P}_2(\bar{\theta}_1) \text{ on } \Gamma_1^D, \quad (3.25b)$$

where $\delta\hat{\theta}_1 = 0$ on Γ_1^D , $\mathbb{P}_2(\bar{\theta}_1)$ is the \mathcal{L}_2 projection of the Dirichlet data into the approximation space on Γ_1^D and $\hat{\mathbf{q}}_1 \cdot \mathbf{n}$ is the HDG numerical normal flux, which is defined as,

$$\hat{\mathbf{q}}_1 \cdot \mathbf{n} = \mathbf{q}_1 \cdot \mathbf{n} + \tau(\theta_1 - \hat{\theta}_1) \text{ on } \partial\Omega_1^e \setminus \Gamma_I. \quad (3.26)$$

Essentially, the global problem (3.25a) states the so-called conservativity condition, *i.e.*, the weak continuity of the normal flux across all the interior faces of the mesh in Ω_1 . The continuity of the fluxes on the interface, Γ_I , *i.e.*, equation (3.17g), is imposed between the numerical normal flux of HDG, $\hat{\mathbf{q}}_1 \cdot \mathbf{n}_1$, which is defined as,

$$\hat{\mathbf{q}}_1 \cdot \mathbf{n}_1 = \mathbf{q}_1 \cdot \mathbf{n}_1 + \tau(\theta_1 - \theta_2) \text{ on } \partial\Omega_1^e \cap \Gamma_I, \quad (3.27)$$

and the flux on the interface from Ω_2 , which is $-\alpha_2 \text{grad } \theta_2 \cdot \mathbf{n}_2$, leading to,

$$-\langle \delta\theta_2, \mathbf{q}_1 \cdot \mathbf{n}_1 + \tau(\theta_1 - \theta_2) \rangle_{\Gamma_I} + \langle \delta\theta_2, \alpha_2 \text{grad } \theta_2 \cdot \mathbf{n}_2 \rangle_{\Gamma_I} = 0. \quad (3.28)$$

By summing equation (3.28) to the weak form of CG (3.22) in Ω_2 , and using the weak formulation of HDG, (3.23) and (3.25), in Ω_1 , the coupled discrete problem is obtained: find $(\mathbf{q}_1, \theta_1, \hat{\theta}_1, \theta_2) \in [\mathcal{V}_k^h(\Omega_1)]^2 \times \mathcal{V}_k^h(\Omega_1) \times \Lambda_k^h(\Gamma_1 \setminus \Gamma_I) \times \mathcal{W}_{k+1}^h(\Omega_2)$ such that $\hat{\theta}_1 = \mathbb{P}_2(\bar{\theta}_1)$ on Γ_1^D , $\theta_2 = \Pi^h(\bar{\theta}_2)$ on Γ_2^D and

$$\begin{aligned} & (\delta\theta_1, \text{div } \mathbf{q}_1)_{\Omega_1^e} + \left\langle \delta\theta_1, \tau(\theta_1 - \hat{\theta}_1) \right\rangle_{\partial\Omega_1^e \setminus \Gamma_I} \\ & \quad + \langle \delta\theta_1, \tau(\theta_1 - \theta_2) \rangle_{\partial\Omega_1^e \cap \Gamma_I} - (\delta\theta_1, \bar{g}_1)_{\Omega_1^e} = 0, \\ & (\delta\mathbf{q}_1, \alpha_1^{-1} \mathbf{q}_1)_{\Omega_1^e} - (\text{div } \delta\mathbf{q}_1, \theta_1)_{\Omega_1^e} + \left\langle \delta\mathbf{q}_1 \cdot \mathbf{n}, \hat{\theta}_1 \right\rangle_{\partial\Omega_1^e \setminus \Gamma_I} \\ & \quad + \langle \delta\mathbf{q}_1 \cdot \mathbf{n}_1, \theta_2 \rangle_{\partial\Omega_1^e \cap \Gamma_I} = 0, \end{aligned} \quad (3.29a)$$

for $e = 1, \dots, m_{el}$ and,

$$\sum_{e=1}^{m_{el}} \left\langle \delta\hat{\theta}_1, \left(\mathbf{q}_1 \cdot \mathbf{n} + \tau(\theta_1 - \hat{\theta}_1) \right) \right\rangle_{\partial\Omega_1^e \setminus \Gamma_I} = 0, \quad (3.29b)$$

$$\begin{aligned} & (\text{grad } \delta\theta_2, \alpha_2 \text{grad } \theta_2)_{\Omega_2} - \langle \delta\theta_2, \mathbf{q}_1 \cdot \mathbf{n}_1 + \tau(\theta_1 - \theta_2) \rangle_{\Gamma_I} \\ & \quad - (\delta\theta_2, \bar{g}_2)_{\Omega_2} = 0, \end{aligned} \quad (3.29c)$$

for all $(\delta\mathbf{q}_1, \delta\theta_1, \delta\hat{\theta}_1, \delta\theta_2) \in [\mathcal{V}_k^h(\Omega_1)]^2 \times \mathcal{V}_k^h(\Omega_1) \times \Lambda_k^h(\Gamma_1 \setminus \Gamma_I) \times \mathcal{W}_{k+1}^h(\Omega_2)$ such that $\delta\hat{\theta}_1 = 0$ on Γ_1^D and $\delta\theta_2 = 0$ on Γ_2^D , where the discrete spaces are defined in (1.2). As usual in HDG, the spaces for approximation in Ω_1 , \mathcal{V}_k^h and Λ_k^h , consider polynomials of the same degree k for all variables. Numerical tests in Section 3.3.6 show that the HDG super-convergence cannot be retained by the coupling with the CG approximation of same degree, k . However, as expected, convergence rates of order $k+2$ for the solution, and of order $k+1$ for the flux, are obtained when higher degree, $k+1$, is considered for the CG domain. Hence, CG approximation space, \mathcal{W}_{k+1}^h , spans polynomials of degree $k+1$.

The discretization of the system of equations in (3.29) gives rise to a matrix equation of the form

$$\begin{bmatrix} \mathbf{A}_{\hat{\theta}\hat{\theta}} & \mathbf{0} & \mathbf{A}_{\hat{\theta}\theta} & \mathbf{A}_{\hat{\theta}q} \\ \mathbf{0} & \mathbf{K}_{\theta\theta} & \mathbf{B}_{\theta\theta}^T & \mathbf{B}_{\theta q} \\ \mathbf{A}_{\theta\hat{\theta}} & \mathbf{B}_{\theta\theta} & \mathbf{A}_{\theta\theta} & \mathbf{A}_{\theta q} \\ \mathbf{A}_{q\hat{\theta}} & \mathbf{B}_{q\theta} & \mathbf{A}_{q\theta} & \mathbf{A}_{qq} \end{bmatrix} \begin{bmatrix} \hat{\theta}_1 \\ \theta_2 \\ \theta_1 \\ \mathbf{q}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{g}}_2 \\ \bar{\mathbf{g}}_1 \\ \mathbf{0} \end{bmatrix}. \quad (3.30)$$

The column vectors $\boldsymbol{\theta}_1$, \mathbf{q}_1 , $\hat{\boldsymbol{\theta}}_1$ and $\boldsymbol{\theta}_2$ contain the DOFs associated to θ_1 , q_1 , $\hat{\theta}_1$ and θ_2 , respectively. Static condensation is assumed for the CG discretization, expressing the nodal values of interior nodes of the element in terms of the nodal values on the edges. The elemental matrices in equation (3.30) are defined in the Appendix C.

As discussed in previous chapters, one of the attractive features of HDG is being able to express the local variables, $\boldsymbol{\theta}_1$ and \mathbf{q}_1 , in terms of the trace variable $\hat{\boldsymbol{\theta}}_1$. This is done by using static condensation technique applied in an element-by-element fashion (see, for instance Sevilla and Huerta (2016)). In the present case, $\hat{\boldsymbol{\theta}}_1$ and $\boldsymbol{\theta}_2$ are coupled at the interface and hence, the local variables, $\boldsymbol{\theta}_1$ and \mathbf{q}_1 , in the HDG domain Ω_1 are expressed in terms of $\hat{\boldsymbol{\theta}}_1$ and $\boldsymbol{\theta}_2$. Therefore, system (3.30) can be expressed as follows,

$$\begin{Bmatrix} \boldsymbol{\theta}_1 \\ \mathbf{q}_1 \end{Bmatrix} = \begin{bmatrix} \mathbf{A}_{\theta\theta} & \mathbf{A}_{\theta q} \\ \mathbf{A}_{q\theta} & \mathbf{A}_{qq} \end{bmatrix}^{-1} \left(\begin{Bmatrix} \bar{\mathbf{g}}_1 \\ \mathbf{0} \end{Bmatrix} - \begin{bmatrix} \mathbf{A}_{\theta\hat{\theta}} & \mathbf{B}_{\theta\theta} \\ \mathbf{A}_{q\hat{\theta}} & \mathbf{B}_{q\theta} \end{bmatrix} \begin{Bmatrix} \hat{\boldsymbol{\theta}}_1 \\ \boldsymbol{\theta}_2 \end{Bmatrix} \right), \quad (3.31a)$$

$$\begin{bmatrix} \mathbf{A}_{\hat{\theta}\theta} & \mathbf{A}_{\hat{\theta}q} \\ \mathbf{B}_{\theta\theta}^T & \mathbf{B}_{\theta q} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\theta}_1 \\ \mathbf{q}_1 \end{Bmatrix} + \begin{bmatrix} \mathbf{A}_{\hat{\theta}\hat{\theta}} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{\theta\theta} \end{bmatrix} \begin{Bmatrix} \hat{\boldsymbol{\theta}}_1 \\ \boldsymbol{\theta}_2 \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \bar{\mathbf{g}}_2 \end{Bmatrix}. \quad (3.31b)$$

Equation (3.31a) is the so-called HDG local solver, which can be computed element-wise owing to the fact that the equations corresponding to the local problem of an element do not involve elemental variables of other elements *i.e.*, the inverted matrix is block diagonal. By replacing equation (3.31a) in (3.31b), $\boldsymbol{\theta}_1$ and \mathbf{q}_1 are eliminated resulting in a system with unknowns only in $\hat{\boldsymbol{\theta}}_1$ and $\boldsymbol{\theta}_2$:

$$\begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{Bmatrix} \hat{\boldsymbol{\theta}}_1 \\ \boldsymbol{\theta}_2 \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{Bmatrix}, \quad (3.32)$$

where

$$\begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{\hat{\theta}\hat{\theta}} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{\theta\theta} \end{bmatrix} - \begin{bmatrix} \mathbf{A}_{\hat{\theta}\theta} & \mathbf{A}_{\hat{\theta}q} \\ \mathbf{B}_{\theta\theta}^T & \mathbf{B}_{\theta q} \end{bmatrix} \begin{bmatrix} \mathbf{A}_{\theta\theta} & \mathbf{A}_{\theta q} \\ \mathbf{A}_{q\theta} & \mathbf{A}_{qq} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{A}_{\theta\hat{\theta}} & \mathbf{B}_{\theta\theta} \\ \mathbf{A}_{q\hat{\theta}} & \mathbf{B}_{q\theta} \end{bmatrix}, \quad (3.33a)$$

$$\begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \bar{\mathbf{g}}_2 \end{Bmatrix} - \begin{bmatrix} \mathbf{A}_{\hat{\theta}\theta} & \mathbf{A}_{\hat{\theta}q} \\ \mathbf{B}_{\theta\theta}^T & \mathbf{B}_{\theta q} \end{bmatrix} \begin{bmatrix} \mathbf{A}_{\theta\theta} & \mathbf{A}_{\theta q} \\ \mathbf{A}_{q\theta} & \mathbf{A}_{qq} \end{bmatrix}^{-1} \begin{Bmatrix} \bar{\mathbf{g}}_1 \\ \mathbf{0} \end{Bmatrix}. \quad (3.33b)$$

After solving the system in (3.32) for $\hat{\boldsymbol{\theta}}_1$ and $\boldsymbol{\theta}_2$, the HDG elemental local variables, $\boldsymbol{\theta}_1$ and \mathbf{q}_1 , and the CG nodal values of the interior nodes can be computed by using equation (3.31a), and the CG static condensation.

3.3.5 Alternative CG-HDG coupled formulation with a projection operator on the interface

The coupled formulation presented earlier considers the standard HDG local problem for the elements that do not share the interface, $\partial\Omega_1^e \cap \Gamma_I = \emptyset$, and a non-standard HDG local problem imposing (3.24) in weak form for elements along the interface, $\partial\Omega_1^e \cap \Gamma_I \neq \emptyset$. In terms of implementation, additional matrices, $\mathbf{B}_{\theta\theta}$ and $\mathbf{B}_{\theta q}$, in the elements along the interface boundary Γ_I are needed for the non-standard HDG local solver. An alternative coupling formulation is proposed in this section to keep the implementation changes to minimum in any existing HDG and CG codes. The main idea in this formulation is to use a projection

to satisfy the transmission conditions. This formulation only requires the standard elemental matrices from HDG (after static condensation of local variables into trace variable) and CG domains, and a projection operation is used on the HDG elemental matrices before assembling into the global system. This requires minimal changes to the existing codes and it is noticed that, in the numerical results, this implementation gives practically the same results as the earlier one with same convergence rates.

In this case, the Dirichlet boundary conditions for the local problem are defined as follows,

$$\theta_1 = \begin{cases} \hat{\theta}_1 & \text{on } \partial\Omega_1^e \setminus \Gamma_I, \\ \mathbb{P}_2(\theta_2) & \text{on } \partial\Omega_1^e \cap \Gamma_I, \end{cases} \quad (3.34)$$

where in equation (3.34) the operator \mathbb{P}_2 stands for the \mathcal{L}_2 projection from the CG space \mathcal{W}^h to the HDG space \mathcal{V}^h . Hence, the trace is set to the projection of the CG solution on the faces along the interface, *i.e.*, $\hat{\theta}_1 = \mathbb{P}_2(\theta_2)$ on $\partial\Omega_1^e \cap \Gamma_I$. Consequently, the numerical normal flux along the interface is defined as,

$$\hat{\mathbf{q}}_1 \cdot \mathbf{n}_1 = \mathbf{q}_1 \cdot \mathbf{n}_1 + \tau(\theta_1 - \mathbb{P}_2(\theta_2)) \quad \text{on } \partial\Omega_1^e \cap \Gamma_I. \quad (3.35)$$

The jump of fluxes along the interface is weighted with $\mathbb{P}_2(\delta\theta_2)$ leading to,

$$-\langle \mathbb{P}_2(\delta\theta_2), \mathbf{q}_1 \cdot \mathbf{n}_1 + \tau(\theta_1 - \mathbb{P}_2(\theta_2)) \rangle_{\Gamma_I} + \langle \mathbb{P}_2(\delta\theta_2), \alpha_2 \text{grad } \theta_2 \cdot \mathbf{n}_2 \rangle_{\Gamma_I} = 0. \quad (3.36)$$

Using (3.34) and (3.36), the weak formulation of the coupled discrete problem becomes: find $(\mathbf{q}_1, \theta_1, \hat{\theta}_1, \theta_2) \in [\mathcal{V}_k^h(\Omega_1)]^2 \times \mathcal{V}_k^h(\Omega_1) \times \Lambda_k^h(\Gamma_1 \setminus \Gamma_I) \times \mathcal{W}_{k+1}^h(\Omega_2)$ such that $\hat{\theta}_1 = \mathbb{P}_2(\bar{\theta}_1)$ on Γ_1^D , $\theta_2 = \Pi^h(\bar{\theta}_2)$ on Γ_2^D and

$$\begin{aligned} (\delta\theta_1, \text{div } \mathbf{q}_1)_{\Omega_1^e} + \langle \delta\theta_1, \tau(\theta_1 - \hat{\theta}_1) \rangle_{\partial\Omega_1^e \setminus \Gamma_I} \\ + \langle \delta\theta_1, \tau(\theta_1 - \mathbb{P}_2(\theta_2)) \rangle_{\partial\Omega_1^e \cap \Gamma_I} - (\delta\theta_1, \bar{g}_1)_{\Omega_1^e} = 0, \\ (\delta\mathbf{q}_1, \alpha_1^{-1} \mathbf{q}_1)_{\Omega_1^e} - (\text{div } \delta\mathbf{q}_1, \theta_1)_{\Omega_1^e} + \langle \delta\mathbf{q}_1 \cdot \mathbf{n}, \hat{\theta}_1 \rangle_{\partial\Omega_1^e \setminus \Gamma_I} \\ + \langle \delta\mathbf{q}_1 \cdot \mathbf{n}_1, \mathbb{P}_2(\theta_2) \rangle_{\partial\Omega_1^e \cap \Gamma_I} = 0, \end{aligned} \quad (3.37a)$$

for $e = 1, \dots, m_{el}$ and,

$$\sum_{e=1}^{m_{el}} \langle \delta\hat{\theta}_1, (\mathbf{q}_1 \cdot \mathbf{n} + \tau(\theta_1 - \hat{\theta}_1)) \rangle_{\partial\Omega_1^e \setminus \Gamma_I} = 0, \quad (3.37b)$$

$$\begin{aligned} (\text{grad } \delta\theta_2, \alpha_2 \text{grad } \theta_2)_{\Omega_2} - \langle \mathbb{P}_2(\delta\theta_2), \mathbf{q}_1 \cdot \mathbf{n}_1 + \tau(\theta_1 - \mathbb{P}_2(\theta_2)) \rangle_{\Gamma_I} \\ + \langle (\mathbb{P}_2(\delta\theta_2) - \delta\theta_2), \alpha_2 \text{grad } \theta_2 \cdot \mathbf{n}_2 \rangle_{\Gamma_I} - (\delta\theta_2, \bar{g}_2)_{\Omega_2} = 0, \end{aligned} \quad (3.37c)$$

for all $(\delta\mathbf{q}_1, \delta\theta_1, \delta\hat{\theta}_1, \delta\theta_2) \in [\mathcal{V}_k^h(\Omega_1)]^2 \times \mathcal{V}_k^h(\Omega_1) \times \Lambda_k^h(\Gamma_1 \setminus \Gamma_I) \times \mathcal{W}_{k+1}^h(\Omega_2)$ such that $\delta\hat{\theta}_1 = 0$ on Γ_1^D and $\delta\theta_2 = 0$ on Γ_2^D .

The weak form in equations (3.37) is similar to one presented earlier in equations (3.29) except for two major differences. First, θ_2 is now replaced by its projection, $\mathbb{P}_2(\theta_2)$, in the HDG local problems (3.37a), and, second, an additional term $\langle (\mathbb{P}_2(\delta\theta_2) - \delta\theta_2), \alpha_2 \text{grad } \theta_2 \cdot \mathbf{n}_2 \rangle_{\Gamma_I}$ appears in the last equation (3.37c). The implementation of this new term maybe cumbersome, because it requires the computation of the gradient of the CG elemental basis functions on the integration points on the interface. However, $\mathbb{P}_2(\delta\theta_2) - \delta\theta_2 = \mathcal{O}(h^{k+1})$, where h is the mesh size and k is degree of approximation, and therefore, this term can be safely neglected in the discrete problem without losing neither the convergence nor the accuracy of the solution.

This formulation does not require the computation of the new matrices that arise in (3.29), namely $\mathbf{B}_{\theta\theta}$ and $\mathbf{B}_{\theta q}$. The projection operation can be done in an element-by-element basis on the HDG elemental matrices for the elements along the interface boundary. For the sake of simplifying the presentation the nodal values of the CG approximation, θ_2 , are split into values for nodes on the interface, θ_2^I , and the remaining CG nodal values, θ_2^i . The system of equations can be then represented as,

$$\begin{bmatrix} \mathbf{A}_{\hat{\theta}\hat{\theta}} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{\hat{\theta}\theta} & \mathbf{A}_{\hat{\theta}q} \\ \mathbf{0} & \mathbf{P}^T \mathbf{A}_{\hat{\theta}\hat{\theta}} \mathbf{P} + \mathbf{K}_{\theta\theta}^{II} & \mathbf{K}_{\theta\theta}^{Ii} & \mathbf{P}^T \mathbf{A}_{\hat{\theta}\theta} & \mathbf{P}^T \mathbf{A}_{\hat{\theta}q} \\ \mathbf{0} & \mathbf{K}_{\theta\theta}^{iI} & \mathbf{K}_{\theta\theta}^{ii} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{A}_{\theta\hat{\theta}} & \mathbf{A}_{\theta\hat{\theta}} \mathbf{P} & \mathbf{0} & \mathbf{A}_{\theta\theta} & \mathbf{A}_{\theta q} \\ \mathbf{A}_{q\hat{\theta}} & \mathbf{A}_{q\hat{\theta}} \mathbf{P} & \mathbf{0} & \mathbf{A}_{q\theta} & \mathbf{A}_{qq} \end{bmatrix} \begin{Bmatrix} \hat{\theta}_1 \\ \theta_2^I \\ \theta_2^i \\ \theta_1 \\ \mathbf{q}_1 \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \bar{\mathbf{g}}_2^I \\ \bar{\mathbf{g}}_2^i \\ \bar{\mathbf{g}}_1 \\ \mathbf{0} \end{Bmatrix}, \quad (3.38)$$

where in equation (3.38), \mathbf{P} is the assembly of projection matrices on all the faces along the interface. This implementation can be easily plugged into any existing HDG solver for heat equation.

Both coupled formulations (3.29) and (3.37) have been implemented and the comparison of the numerical results inferred that both are practically identical. In some tests, the first proposed formulation (3.29) gave slightly smaller errors. The difference might be due to neglecting the term $\mathbb{P}_2(\delta\theta_2) - \delta\theta_2$ in the last formulation (3.37). However, the difference in the errors — even for the coarsest mesh — is negligible and, hence, in all the results presented in this work, the formulation with projection (3.37) is used neglecting the term $\mathbb{P}_2(\delta\theta_2) - \delta\theta_2$.

3.3.6 Convergence results

In this section, the convergence results of the coupled CG-HDG formulation for the heat equation are presented. A square domain, $\Omega := [0, 1]^2$ is considered with the analytical solution,

$$\theta = 1 + \cos(\pi x_1) \cos(\pi x_2). \quad (3.39)$$

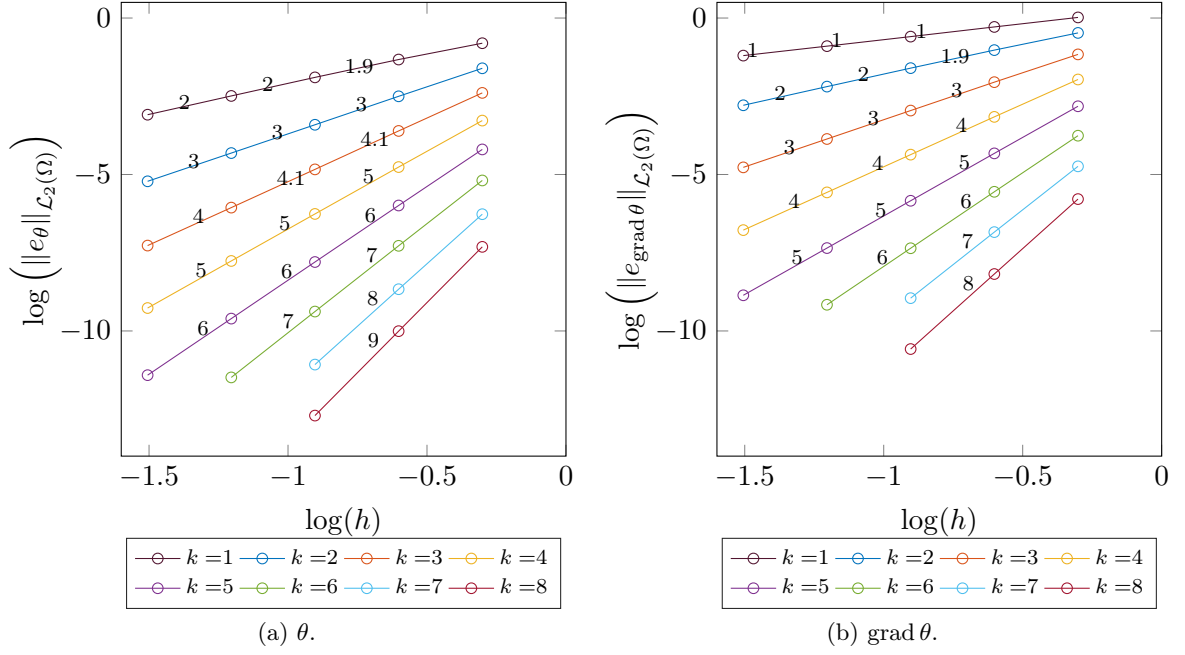
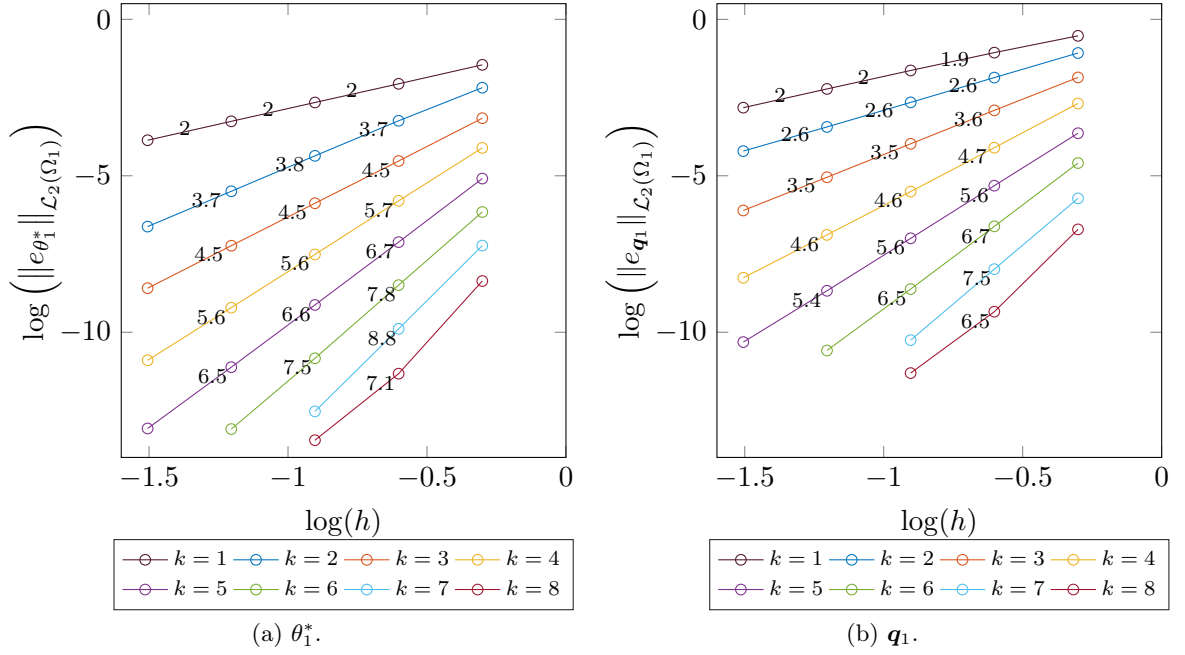
Dirichlet boundary conditions are prescribed on all the exterior boundary.

The domain is divided into two halves in vertical direction. The domain corresponding to HDG is $\Omega_1 := [0, 0.5] \times [0, 1]$, the CG domain is $\Omega_2 := [0.5, 1] \times [0, 1]$, and the interface, Γ_I , is $x_1 = 0.5$. A suitable body force is computed from the heat equation with the considered analytical solution for both domains, with the diffusivity constants, $\alpha_2 = \alpha_1 = 1$. The results presented here consider a stability parameter of $\tau = 1$ on all faces of each element in HDG domain, Ω_1 .

Figure 3.8 shows the convergence for the coupled formulation with same degree for CG and HDG, with $k = 1 - 8$ and element size, $h = 1/\{2^1, 2^2, 2^3, 2^4, 2^5\}$. The error in θ is measured by using \mathcal{L}_2 norm of errors of the post-processed solution, θ_1^* , in Ω_1 , and CG solution, θ_2 , in Ω_2 . Similarly, error in $\text{grad } \theta$ is computed using \mathcal{L}_2 norm of error in \mathbf{q}_1 in Ω_1 and error in $\text{grad } \theta_2$ in Ω_2 , that is,

$$\begin{aligned} \|e_\theta\|_{\mathcal{L}_2(\Omega)} &= \sqrt{\|e_{\theta_1^*}\|_{\mathcal{L}_2(\Omega_1)}^2 + \|e_{\theta_2}\|_{\mathcal{L}_2(\Omega_2)}^2}, \\ \|e_{\text{grad } \theta}\|_{\mathcal{L}_2(\Omega)} &= \sqrt{\|e_{\mathbf{q}_1}\|_{\mathcal{L}_2(\Omega_1)}^2 + \|e_{\text{grad } \theta_2}\|_{\mathcal{L}_2(\Omega_2)}^2}. \end{aligned} \quad (3.40)$$

When the degree of approximation k is used for both HDG and CG domains, even though HDG has superior convergence properties, errors in CG domain dominates for both θ and

Figure 3.8: Coupled CG(T_k)-HDG(T_k): convergence plots in Ω .Figure 3.9: Coupled CG(T_k)-HDG(T_k): convergence plots in Ω_1 .

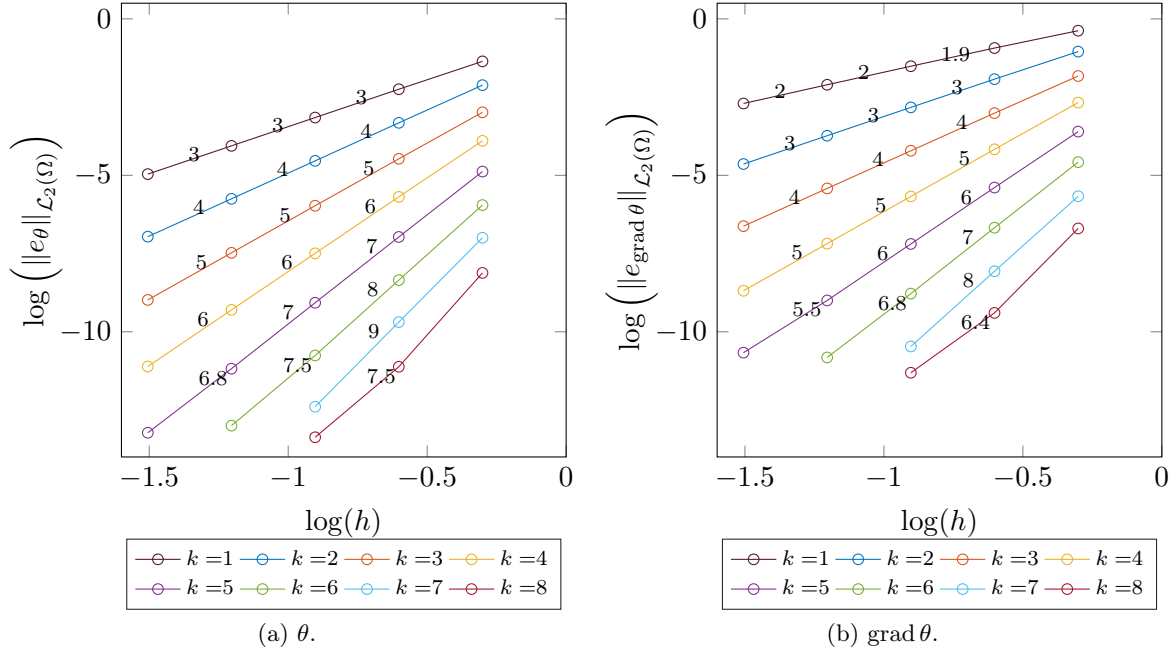


Figure 3.10: Coupled CG(T_{k+1})-HDG(T_k): convergence plots in Ω .

$\text{grad } \theta$. Hence, as shown in fig. 3.8a, the order of convergence of the coupled solution is $k + 1$ for θ . Similarly, for $\text{grad } \theta$, the order of convergence is k .

Figure 3.9 shows the convergence of θ_1^* and \mathbf{q}_1 in Ω_1 for the coupled CG(T_k)-HDG(T_k) model. Sub-optimal convergence rates are observed in both variables: for $k > 2$, θ_1^* converges with only $k + 1.5$ instead of $k + 2$, and \mathbf{q}_1 converges with order $k + 0.5$ instead of $k + 1$. HDG super-convergence requires a solution of order $k + 1$ for \mathbf{q}_1 , and mean of θ_1 that converges with order $k + 2$ in each element in Ω_1 . The elements in Ω_1 that share the interface, Γ_I , do not possess the mentioned convergence rates because of the coupling with CG domain, Ω_2 .

To address this shortcoming, higher degree of approximation is considered for the CG discretization. Figure 3.10a shows the convergence plots for a coupled approximation with degree k for HDG and degree $k + 1$ for CG. Optimal convergence rates of both methods are retained in this case. The post-processed solution of HDG with degree k has the same order of convergence, which is $k + 2$, as the CG solution with degree $k + 1$. Similarly, the flux \mathbf{q}_1 of HDG converges with same order as $\text{grad } \theta_2$ of CG, which is $k + 1$.

The same conclusions are drawn for quadrilateral elements as well and, hence, the results are omitted.

3.4 Coupled CG-HDG formulation for conjugate heat transfer problem

3.4.1 Governing equations

Following fig. 3.6, the governing equations of the conjugate heat transfer problem in Ω_1 and Ω_2 are expressed as follows,

$$\begin{aligned} \mathbf{L} - \text{grad } \mathbf{u} &= \mathbf{0} & \text{in } \Omega_1, \\ \text{div}(\mathbf{u} \otimes \mathbf{u}) - \text{div}(-p\mathbf{I} + \nu \mathbf{L}) - \mathbf{f}(\theta_1) &= \bar{\mathbf{s}} & \text{in } \Omega_1, \\ \text{div } \mathbf{u} &= 0 & \text{in } \Omega_1, \end{aligned} \quad (3.41a)$$

$$\begin{aligned} \mathbf{q}_1 + \alpha_1 \text{grad } \theta_1 &= \mathbf{0} & \text{in } \Omega_1, \\ \text{div}(\mathbf{q}_1 + \mathbf{u} \theta_1) &= \bar{g}_1 & \text{in } \Omega_1, \end{aligned} \quad (3.41b)$$

$$-\text{div}(\alpha_2 \text{grad } \theta_2) = \bar{g}_2 \quad \text{in } \Omega_2, \quad (3.41c)$$

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } \Gamma_1^D \cup \Gamma_1^N \cup \Gamma_I, \quad (3.41d)$$

$$\theta_1 = \bar{\theta}_1 \quad \text{on } \Gamma_1^D, \quad (3.41e)$$

$$(\mathbf{q}_1 + \mathbf{u} \theta_1) \cdot \mathbf{n} = \bar{q}_{n1} \quad \text{on } \Gamma_1^N,$$

$$\theta_2 = \bar{\theta}_2 \quad \text{on } \Gamma_2^D, \quad (3.41f)$$

$$(-\alpha_2 \text{grad } \theta_2) \cdot \mathbf{n} = \bar{q}_{n2} \quad \text{on } \Gamma_2^N,$$

$$\theta_1 - \theta_2 = 0 \quad \text{on } \Gamma_I, \quad (3.41g)$$

$$\mathbf{q}_1 \cdot \mathbf{n}_1 - (\alpha_2 \text{grad } \theta_2) \cdot \mathbf{n}_2 = 0 \quad \text{on } \Gamma_I.$$

The body force, $\bar{\mathbf{s}}$, prescribed velocity, $\bar{\mathbf{u}}$, and heat generation, \bar{g}_2 and \bar{g}_1 , are zero for the conjugate heat transfer problem, *per se*. However, they are included in the governing equations as they have non-zero contribution in the numerical experiments considered in the convergence results. In the case of conjugate heat transfer problem, no-slip boundary condition is applied for fluid on all the boundary including the interface. Therefore, convective flux entering the solid domain is zero and the transmission conditions presented in (3.17f) and (3.17g) are valid for this example. In the numerical tests of convergence analysis, the convective flux on the interface is non-zero due to non-zero prescribed boundary velocity. However, this convective flux is assumed to be zero in order to use the same transmission conditions in (3.17g). The extension of coupled formulation presented in Section 3.3 to the conjugate heat transfer problem is straightforward and it is presented in next section.

3.4.2 Weak formulation

The coupled CG-HDG formulation presented in Section 3.3.5 is used. Using (3.37) and (3.5), the weak formulation for the conjugate heat transfer problem can be stated as: find $(\mathbf{u}, p, \mathbf{L}, \theta_1, \mathbf{q}_1, \hat{\mathbf{u}}, \rho_e, \hat{\theta}_1, \theta_2) \in [\mathcal{V}_k^h(\Omega_1)]^2 \times \mathcal{V}_k^h(\Omega_1) \times [\mathcal{V}_k^h(\Omega_1)]^{2 \times 2} \times \mathcal{V}_k^h(\Omega_1) \times [\mathcal{V}_k^h(\Omega_1)]^2 \times [\Lambda_k^h(\Gamma_1 \setminus \Gamma_I)]^2 \times \mathbb{R}^{m_{el}} \times \Lambda_k^h(\Gamma_1 \setminus \Gamma_I) \times \mathcal{W}_{k+1}^h(\Omega_2)$ such that $\mathbf{u} = \Pi^h(\bar{\mathbf{u}})$ on $\Gamma_1^D \cup \Gamma_1^N \cup \Gamma_I$, $\hat{\theta}_1 = \mathbb{P}_2(\bar{\theta}_1)$ on Γ_1^D , $\theta_2 = \Pi^h(\bar{\theta}_2)$ on Γ_2^D and

$$\begin{aligned} (\delta \mathbf{L}, \mathbf{L})_{\Omega_1^e} + (\text{div } \delta \mathbf{L}, \mathbf{u})_{\Omega_1^e} - \langle \delta \mathbf{L} \mathbf{n}, \hat{\mathbf{u}} \rangle_{\partial \Omega_1^e} &= 0, \\ -(\text{grad } \delta \mathbf{u}, \mathbf{u} \otimes \mathbf{u})_{\Omega_1^e} - (\delta \mathbf{u}, \text{div}(-p\mathbf{I} + \nu \mathbf{L}))_{\Omega_1^e} \\ + \langle \delta \mathbf{u}, (\hat{\mathbf{u}} \otimes \hat{\mathbf{u}}) \mathbf{n} + \tau_u (\mathbf{u} - \hat{\mathbf{u}}) \rangle_{\partial \Omega_1^e} - (\delta \mathbf{u}, \mathbf{f}(\theta_1))_{\Omega_1^e} - (\delta \mathbf{u}, \bar{\mathbf{s}})_{\Omega_1^e} &= 0, \\ -(\text{grad } \delta p, \mathbf{u})_{\Omega_1^e} + \langle \delta p, \hat{\mathbf{u}} \cdot \mathbf{n} \rangle_{\partial \Omega_1^e} &= 0, \end{aligned} \quad (3.42a)$$

$$\frac{1}{|\partial\Omega_1^e|} \langle p, 1 \rangle_{\partial\Omega_1^e} = \rho_e, \quad (3.42b)$$

$$\begin{aligned} & (\delta\theta_1, \operatorname{div} \mathbf{q}_1)_{\Omega_1^e} - (\operatorname{grad} \delta\theta_1, \mathbf{u} \theta_1)_{\Omega_1^e} + \langle \delta\theta_1, (\hat{\mathbf{u}} \cdot \mathbf{n}) \hat{\theta}_1 \rangle_{\partial\Omega_1^e \setminus \Gamma_I} \\ & + \langle \delta\theta_1, \tau_\theta(\theta_1 - \hat{\theta}_1) \rangle_{\partial\Omega_1^e \setminus \Gamma_I} + \langle \delta\theta_1, \tau_\theta(\theta_1 - \mathbb{P}_2(\theta_2)) \rangle_{\partial\Omega_1^e \cap \Gamma_I} - (\delta\theta_1, \bar{g}_1)_{\Omega_1^e} = 0, \\ & (\delta\mathbf{q}_1, \alpha_1^{-1} \mathbf{q}_1)_{\Omega_1^e} - (\operatorname{div} \delta\mathbf{q}_1, \theta_1)_{\Omega_1^e} + \langle \delta\mathbf{q}_1 \cdot \mathbf{n}, \hat{\theta}_1 \rangle_{\partial\Omega_1^e \setminus \Gamma_I} \\ & + \langle \delta\mathbf{q}_1 \cdot \mathbf{n}_1, \mathbb{P}_2(\theta_2) \rangle_{\partial\Omega_1^e \cap \Gamma_I} = 0, \end{aligned} \quad (3.42c)$$

for $e = 1, \dots, m_{el}$, and,

$$\sum_{e=1}^{m_{el}} \langle \delta\hat{\mathbf{u}}, (-p\mathbf{I} + \nu\mathbf{L}) \mathbf{n} + \tau_u(\hat{\mathbf{u}} - \mathbf{u}) \rangle_{\partial\Omega_1^e} = 0, \quad (3.43a)$$

$$\langle \hat{\mathbf{u}} \cdot \mathbf{n}, 1 \rangle_{\partial\Omega_1^e} = 0 \quad \text{for } e = 1, \dots, m_{el},$$

$$\sum_{e=1}^{m_{el}} \left\langle \delta\hat{\theta}_1, \left((\mathbf{q}_1 + \hat{\mathbf{u}} \hat{\theta}_1) \cdot \mathbf{n} + \tau_\theta(\theta_1 - \hat{\theta}_1) \right) \right\rangle_{\partial\Omega_1^e \setminus \Gamma_I} = \left\langle \delta\hat{\theta}_1, \bar{q}_{n1} \right\rangle_{\Gamma_1^N}, \quad (3.43b)$$

$$\begin{aligned} & (\operatorname{grad} \delta\theta_2, \alpha_2 \operatorname{grad} \theta_2)_{\Omega_2} - \langle \mathbb{P}_2(\delta\theta_2), \mathbf{q}_1 \cdot \mathbf{n}_1 + \tau_\theta(\theta_1 - \mathbb{P}_2(\theta_2)) \rangle_{\Gamma_I} \\ & + \langle (\mathbb{P}_2(\delta\theta_2) - \delta\theta_2), \alpha_2 \operatorname{grad} \theta_2 \cdot \mathbf{n}_2 \rangle_{\Gamma_I} - (\delta\theta_2, \bar{g}_2)_{\Omega_2} - \langle \delta\theta_2, \bar{q}_{n2} \rangle_{\Gamma_2^N} = 0. \end{aligned} \quad (3.43c)$$

for all $(\delta\mathbf{u}, \delta p, \delta\mathbf{L}, \delta\theta_1, \delta\mathbf{q}_1, \delta\hat{\mathbf{u}}, \delta\hat{\theta}_1, \delta\theta_2) \in [\mathcal{V}_k^h(\Omega_1)]^2 \times \mathcal{V}_k^h(\Omega_1) \times [\mathcal{V}_k^h(\Omega_1)]^{2 \times 2} \times \mathcal{V}_k^h(\Omega_1) \times [\mathcal{V}_k^h(\Omega_1)]^2 \times [\Lambda_k^h(\Gamma_1 \setminus \Gamma_I)]^2 \times \Lambda_k^h(\Gamma_1 \setminus \Gamma_I) \times \mathcal{W}_{k+1}^h(\Omega_2)$ such that $\delta\hat{\mathbf{u}} = \mathbf{0}$ on $\Gamma_1^D \cup \Gamma_1^N \cup \Gamma_I$, $\delta\hat{\theta}_1 = 0$ on Γ_1^D and $\delta\theta_2 = 0$ on Γ_2^D , where the discrete spaces are defined in (1.2).

Here the coupling is made between heat equation in Ω_2 and the diffusive flux term of the convection-diffusion equation in Ω_1 . As discussed in the Section 3.3.5, the term $\langle (\mathbb{P}_2(\delta\theta_2) - \delta\theta_2), \alpha_2 \operatorname{grad} \theta_2 \cdot \mathbf{n}_2 \rangle_{\Gamma_I}$ can be safely neglected without loss of neither the convergence nor accuracy. Finally, the residual and Jacobian matrices of the weak formulation (3.42) and (3.43) are presented in (3.44) and (3.45), respectively. In order to express the system in a compact form, the non-linear terms are absorbed into linear terms in (3.44) and (3.45). The matrix $\mathbf{A}_{u\theta}$ couples temperature to the momentum equation and the matrices $\mathbf{A}_{\theta\hat{\theta}}(\hat{\mathbf{u}})$, $\mathbf{A}_{\theta\theta}(\mathbf{u})$ couples velocity from Navier–Stokes to convection-diffusion equation. Following the static condensation technique, only unknowns in $\hat{\mathbf{u}}, \mathbf{p}, \hat{\theta}_1, \theta_2^I$ and θ_2^i are solved in the final system and local variables are computed an element-by-element fashion. Recall from Section 3.3.5, θ_2^I and θ_2^i are the DOFs on the interface, Γ , and the interior of domain, Ω_2 .

3.4.3 Convergence results

The convergence results of coupled CG-HDG formulation for conjugate heat transfer problem is presented in this section. The fluid domain is, $\Omega_1 := [0, 2] \times [-0.5, 1.5]$, and the solid domain is, $\Omega_2 := [2, 4] \times [-0.5, 1.5]$. Hence, the interface, Γ_I , lies at $x_1 = 2$. The analytical solutions considered for velocity, pressure and temperature are same as considered in (3.12). Dirichlet boundary conditions are prescribed on all the boundary of Ω_1 for velocity whereas, they are prescribed only for exterior boundaries of Ω_1 and Ω_2 for temperature. The body force and heat generation are computed from the analytical solutions using governing equations. All the material parameters used are same as the ones used in Section 3.2.3. The degree of approximation $k = 1 - 8$ and element size $h = 2/\{2, 2^2, 2^3, 2^4, 2^5, 2^6\}$ in each sub-domain are used in the analysis.

Figure 3.11 shows the convergence of post-processed solution of velocity in Ω_1 and temperature in $\Omega = \Omega_1 \cup \Omega_2$ for triangular elements. It is clear from the plot that the optimal

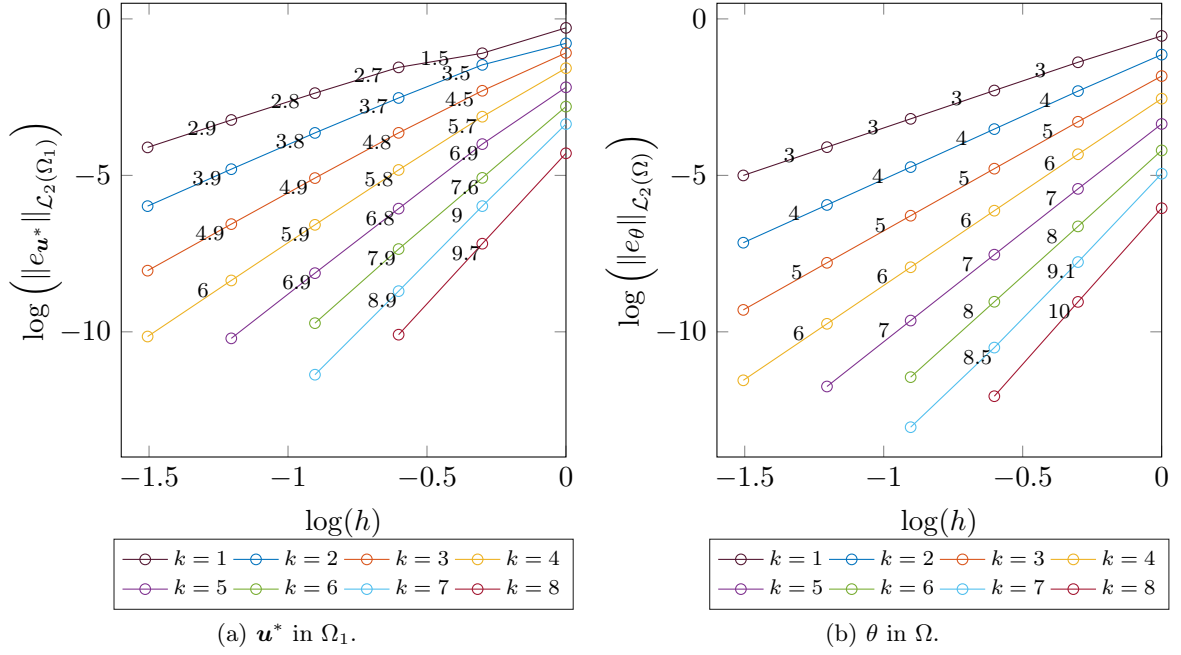


Figure 3.11: Coupled $\text{CG}(T_{k+1})$ -HDG(T_k) for conjugate heat transfer problem: convergence plots of velocity and temperature.

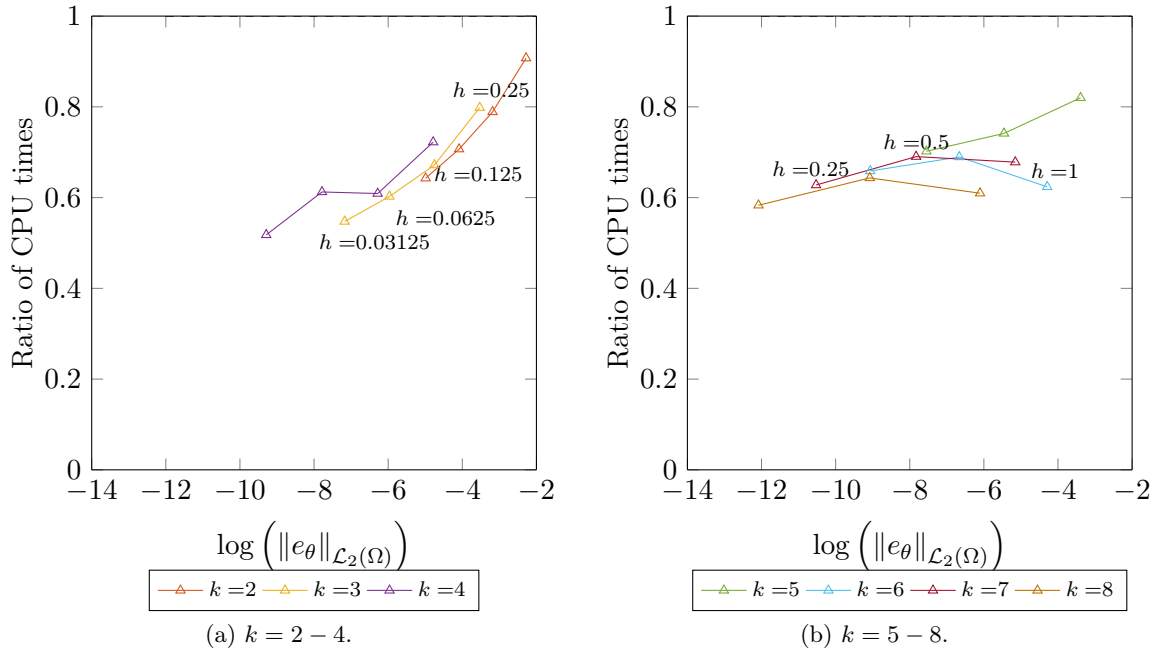


Figure 3.12: Coupled $\text{CG}(T_{k+1})$ -HDG(T_k) for conjugate heat transfer problem: ratio of CPU times for linear solver *vs.* error in temperature for triangular elements. Ratio of CPU times = $\text{CG}(T_{k+1})$ -HDG(T_k) time / $\text{CG}(T_{k+1})$ -CG($T_{k+1}T_k$) time.

$$\begin{bmatrix}
\mathbf{A}_{\hat{u}\hat{u}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{\hat{u}u} & \mathbf{A}_{\hat{u}L} & \mathbf{A}_{\hat{u}p} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{A}_{\hat{p}\hat{u}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{A}_{\hat{\theta}\hat{\theta}}(\hat{\mathbf{u}}) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{\hat{\theta}\theta} & \mathbf{A}_{\hat{\theta}q} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}^T \mathbf{A}_{\hat{\theta}\hat{\theta}}(\hat{\mathbf{u}}) \mathbf{P} + \mathbf{K}_{\theta\theta}^{II} & \mathbf{K}_{\theta\theta}^{Ii} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}^T \mathbf{A}_{\hat{\theta}\theta} & \mathbf{P}^T \mathbf{A}_{\hat{\theta}q} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{K}_{\theta\theta}^{iI} & \mathbf{K}_{\theta\theta}^{ii} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\hline
\mathbf{A}_{u\hat{u}}(\hat{\mathbf{u}}) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{uu}(\mathbf{u}) & \mathbf{A}_{uL} & \mathbf{A}_{up} & \mathbf{0} & \mathbf{A}_{u\theta} & \mathbf{0} \\
\mathbf{A}_{L\hat{u}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{Lu} & \mathbf{A}_{LL} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{A}_{p\hat{u}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{pu} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{pp}^T & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & -\mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{pp} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{A}_{\theta\hat{\theta}}(\hat{\mathbf{u}}) & \mathbf{A}_{\theta\hat{\theta}}(\hat{\mathbf{u}}) \mathbf{P} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{\theta\theta}(\mathbf{u}) & \mathbf{A}_{\theta q} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{A}_{q\hat{\theta}} & \mathbf{A}_{q\hat{\theta}} \mathbf{P} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{q\theta} & \mathbf{A}_{qq} & \mathbf{0}
\end{bmatrix}
\begin{bmatrix}
\hat{\mathbf{u}} \\
\mathbf{p} \\
\hat{\boldsymbol{\theta}}_1 \\
\boldsymbol{\theta}_2^I \\
\boldsymbol{\theta}_2^i \\
\mathbf{u} \\
\mathbf{L} \\
\mathbf{p} \\
\boldsymbol{\lambda} \\
\boldsymbol{\theta}_1 \\
\mathbf{q}_1
\end{bmatrix}
-
\begin{bmatrix}
\bar{\mathbf{t}} \\
\mathbf{0} \\
\bar{\mathbf{q}}_{n1} \\
\bar{\mathbf{g}}_2^I + \bar{\mathbf{q}}_{n2}^I \\
\bar{\mathbf{g}}_2^i + \bar{\mathbf{q}}_{n2}^i \\
\mathbf{f}_{\theta_0} \\
\mathbf{0} \\
\mathbf{0} \\
\mathbf{0} \\
\mathbf{0} \\
\bar{\mathbf{g}}_1 \\
\mathbf{0}
\end{bmatrix}
= \mathbf{0}, \quad (3.44)$$

$$\begin{bmatrix}
\mathbf{A}_{\hat{u}\hat{u}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{A}_{\hat{p}\hat{u}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{A}_{\hat{\theta}\hat{\theta}T}(\hat{\boldsymbol{\theta}}_1) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{\hat{\theta}\theta} & \mathbf{A}_{\hat{\theta}q} \\
\mathbf{P}^T \mathbf{A}_{\hat{\theta}\hat{\theta}T}(\boldsymbol{\theta}_2^I) \mathbf{P} & \mathbf{0} & \mathbf{0} & \mathbf{P}^T \mathbf{A}_{\hat{\theta}\hat{\theta}}(\hat{\mathbf{u}}) \mathbf{P} & \mathbf{K}_{\theta\theta}^{Ii} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}^T \mathbf{A}_{\hat{\theta}\theta} & \mathbf{P}^T \mathbf{A}_{\hat{\theta}q} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{K}_{\theta\theta}^{II} & \mathbf{K}_{\theta\theta}^{ii} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\hline
\mathbf{A}_{u\hat{u}}(\hat{\mathbf{u}}) + \mathbf{A}_{u\hat{u}T}(\hat{\mathbf{u}}) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{uu}(\mathbf{u}) & \mathbf{A}_{uL} & \mathbf{A}_{up} & \mathbf{0} & \mathbf{A}_{u\theta} & \mathbf{0} \\
\mathbf{A}_{L\hat{u}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{Lu} & \mathbf{A}_{LL} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{A}_{p\hat{u}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{pu} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{pp}^T & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & -\mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{pp} & \mathbf{0} & \mathbf{0} \\
\mathbf{A}_{\theta\hat{\theta}T}(\hat{\boldsymbol{\theta}}_1) + \mathbf{A}_{\theta\hat{\theta}T}(\boldsymbol{\theta}_2^I) \mathbf{P} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{\theta\hat{\theta}}(\hat{\mathbf{u}}) & \mathbf{A}_{\theta\hat{\theta}}(\hat{\mathbf{u}}) \mathbf{P} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{\theta\theta}(\mathbf{u}) & \mathbf{A}_{\theta q} & \mathbf{0} \\
\mathbf{A}_{q\hat{\theta}T}(\hat{\boldsymbol{\theta}}_1) + \mathbf{A}_{q\hat{\theta}T}(\boldsymbol{\theta}_2^I) \mathbf{P} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{q\hat{\theta}} & \mathbf{A}_{q\hat{\theta}} \mathbf{P} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{q\theta} & \mathbf{A}_{qq} & \mathbf{0}
\end{bmatrix}
\begin{bmatrix}
\delta \hat{\mathbf{u}} \\
\delta \mathbf{p} \\
\delta \hat{\boldsymbol{\theta}}_1 \\
\delta \boldsymbol{\theta}_2^I \\
\delta \boldsymbol{\theta}_2^i \\
\delta \mathbf{u} \\
\delta \mathbf{L} \\
\delta \mathbf{p} \\
\delta \boldsymbol{\lambda} \\
\delta \boldsymbol{\theta}_1 \\
\delta \mathbf{q}_1
\end{bmatrix}
= -
\begin{bmatrix}
\mathbf{r}_{\hat{u}} \\
\mathbf{r}_{\rho} \\
\mathbf{r}_{\hat{\theta}_1} \\
\mathbf{r}_{\theta_2^I} \\
\mathbf{r}_{\theta_2^i} \\
\mathbf{r}_u \\
\mathbf{r}_L \\
\mathbf{r}_p \\
\mathbf{r}_{\lambda} \\
\mathbf{r}_{\theta_1} \\
\mathbf{r}_{q_1}
\end{bmatrix}. \quad (3.45)$$

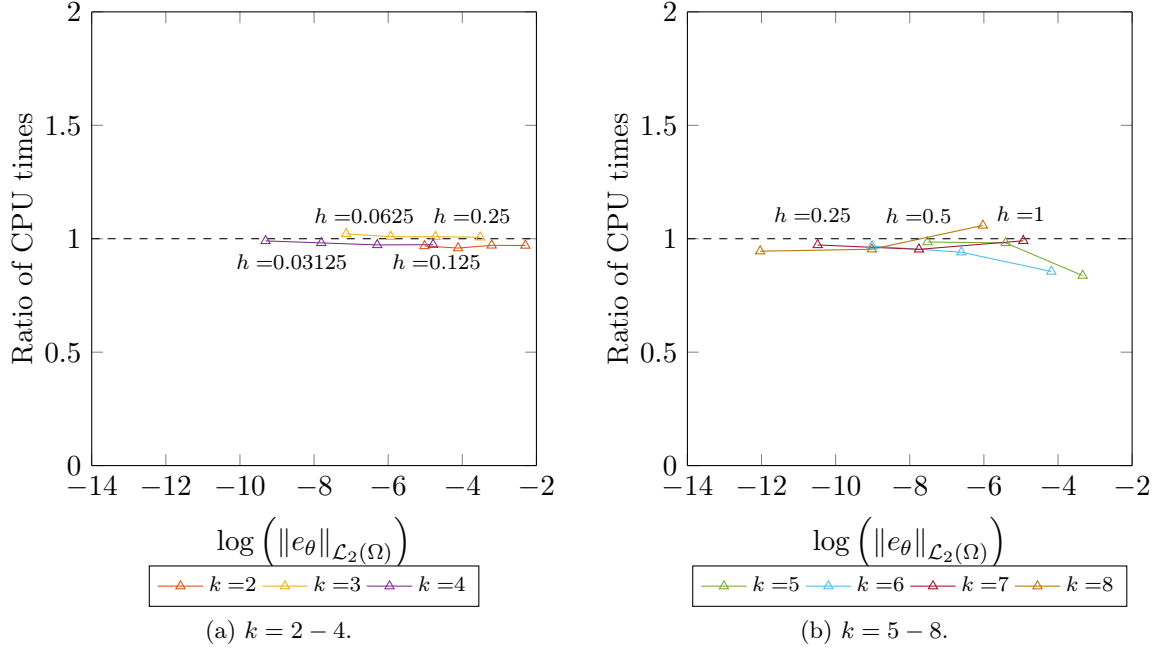


Figure 3.13: Coupled $\text{CG}(T_{k+1})$ -HDG(T_k) for conjugate heat transfer problem: ratio of CPU times for linear solver *vs.* error in temperature for triangular elements. Ratio of CPU times = $\text{CG}(T_{k+1})$ -HDG(T_k) time/HDG(T_k)-HDG(T_k) time.

rate of $k+2$ for the present coupled $\text{CG}(T_{k+1})$ -HDG(T_k) model is obtained for both velocity and temperature. Now, the computational efficiency of the proposed coupled model is considered. The coupled $\text{CG}(T_{k+1})$ -HDG(T_k) model is compared with $\text{CG}(T_{k+1})$ -CG($T_{k+1}T_k$) model, where all the governing equations are discretized by CG method, to study the relative performance in CPU times for linear solver. Figure 3.12 presents the results of the ratio of CPU times for linear solver with error in the temperature in the domain, Ω . As explained earlier in Section 2.6, the errors from the CG-CG model are used and CPU times of CG-HDG model are interpolated corresponding to errors from CG-CG model. All the ratios of the CPU times are less than 1 inferring that coupled CG-HDG model is computationally more efficient than CG-CG model for a given level of accuracy.

The proposed $\text{CG}(T_{k+1})$ -HDG(T_k) model is also compared to HDG(T_k)-HDG(T_k) model to provide a complete discussion on the computational efficiency. In HDG(T_k)-HDG(T_k) model, both fluid and solid domains are discretized with HDG with a degree of approximation k . Both coupled $\text{CG}(T_{k+1})$ -HDG(T_k) and HDG(T_k)-HDG(T_k) models have similar optimal rates of convergence, *i.e.*, $k+2$ for velocity, temperature and $k+1$ for pressure, gradient of velocity and heat flux. Figure 3.13 shows the ratio of CPU times for linear solver between $\text{CG}(T_{k+1})$ -HDG(T_k) and HDG(T_k)-HDG(T_k) CPU times along with the error. The CPU times of $\text{CG}(T_{k+1})$ -HDG(T_k) model is interpolated based on the errors of HDG(T_k)-HDG(T_k) model. Hence, the error in the plot corresponds to the error of post-processed solution of temperature in HDG(T_k)-HDG(T_k) model. It is noticed that the errors in temperature obtained in both HDG(T_k)-HDG(T_k) and $\text{CG}(T_{k+1})$ -HDG(T_k) models are very similar so as the CPU times. It can be noticed from the plot that all the ratios are very close to 1 suggesting that both models exhibits similar computational efficiency for a given level of accuracy.

A theoretical count on number of non-zeros (**nnz**) is presented for $\text{CG}(T_{k+1})$ -CG($T_{k+1}T_k$),

Table 3.5: Expressions of **nnz** in terms of degree of approximation, k , for the considered coupled models. Symbols n_{el}^f and n_{el}^s corresponds to number of elements in fluid and solid domains, respectively.

Tri	CG(T_{k+1})-CG($T_{k+1}T_k$)	$(210k^2 + 246k + 41)n_{el}^f/2 + (15k^2 + 24k + 7)n_{el}^s/2$
	CG(T_{k+1})-HDG(T_k)	$(135k^2 + 294k + 159)n_{el}^f/2 + (15k^2 + 24k + 7)n_{el}^s/2$
	HDG(T_k)-HDG(T_k)	$(135k^2 + 294k + 159)n_{el}^f/2 + (15k^2 + 30k + 30)n_{el}^s/2$
Qua	CG(T_{k+1})-CG($T_{k+1}T_k$)	$(196k^2 + 252k + 68)n_{el}^f + (14k^2 + 24k + 9)n_{el}^s$
	CG(T_{k+1})-HDG(T_k)	$(126k^2 + 268k + 142)n_{el}^f + (14k^2 + 24k + 9)n_{el}^s$
	HDG(T_k)-HDG(T_k)	$(126k^2 + 268k + 142)n_{el}^f + (14k^2 + 28k + 14)n_{el}^s$
Tetra	CG(T_{k+1})-CG($T_{k+1}T_k$)	$(483k^4 + 1182k^3 + 1484k^2 + 528k + 408)n_{el}^f/6$ $+ (21k^4 + 66k^3 + 88k^2 + 40k + 15)n_{el}^s/6$
	CG(T_{k+1})-HDG(T_k)	$(56k^4 + 336k^3 + 740k^2 + 696k + 248)n_{el}^f$ $+ (21k^4 + 66k^3 + 88k^2 + 40k + 15)n_{el}^s/6$
	HDG(T_k)-HDG(T_k)	$(56k^4 + 336k^3 + 740k^2 + 696k + 248)n_{el}^f$ $+ (21k^4 + 126k^3 + 273k^2 + 252k + 84)n_{el}^s/6$
Hexa	CG(T_{k+1})-CG($T_{k+1}T_k$)	$(759k^4 + 2232k^3 + 2889k^2 + 1704k + 489)n_{el}^f$ $+ (33k^4 + 120k^3 + 171k^2 + 108k + 27)n_{el}^s$
	CG(T_{k+1})-HDG(T_k)	$(528k^4 + 2112k^3 + 3204k^2 + 2184k + 564)n_{el}^f$ $+ (33k^4 + 120k^3 + 171k^2 + 108k + 27)n_{el}^s$
	HDG(T_k)-HDG(T_k)	$(528k^4 + 2112k^3 + 3204k^2 + 2184k + 564)n_{el}^f$ $+ (33k^4 + 132k^3 + 198k^2 + 132k + 33)n_{el}^s$

CG(T_{k+1})-HDG(T_k) and HDG(T_k)-HDG(T_k) coupled models. The number of non-zeros in the final stiffness matrix is important as the memory required by the code is directly proportional to it. The problems considered in the present work need relatively smaller computational resources, however, when big computational domains are considered in 3-D, memory requirements can be a bottleneck. The hypothesis presented by Huerta *et al.* (2013) is used in deriving the expressions for **nnz** for the coupled models and it is discussed in detail in Appendix B. The theoretical expressions for **nnz** are presented in table 3.5.

Figures 3.14 and 3.15 presents the ratio of **nnz** between CG-HDG, CG-CG and CG-HDG, HDG-HDG coupled models, respectively. The plots are made assuming that the number of elements in fluid and solid domains are equal. The proposed CG-HDG coupled model has lesser **nnz** in the final stiffness matrix compared to CG-CG model when $k > 1$ for both triangular and quadrilateral elements in 2-D, whereas the ratio goes to below 1 only when $k = 9$ for tetrahedra in 3-D. On the other hand, the ratios are almost constant and close to 1 between $k = 1 - 10$ when CG-HDG and HDG-HDG models are considered in both 2-D and 3-D. Hence, it can be inferred that CG-HDG model has superior computational efficiency and lower memory requirements compared to CG-CG model. However, CG-HDG and HDG-HDG models are very close and almost equivalent in terms of accuracy, efficiency and number of

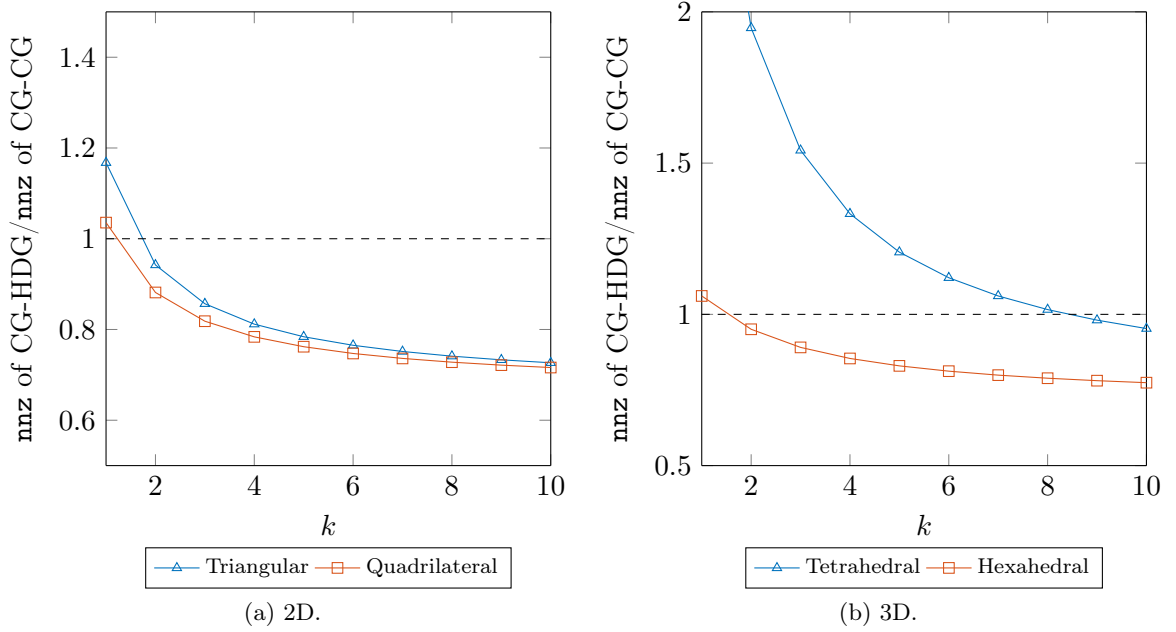


Figure 3.14: Comparison of nnz between coupled $\text{CG}(T_{k+1})\text{-HDG}(T_k)$ and $\text{CG}(T_{k+1})\text{-CG}(T_{k+1}T_k)$ models.

non-zeros. Therefore, in the results presented in the following sections, only CG-CG and CG-HDG models are considered and compared.

3.4.4 Conjugate heat transfer problem

In this example, the proposed models are combined together to solve a benchmark conjugate heat transfer problem. The results of proposed coupled CG-HDG model are presented where the fluid part, which is governed by Navier–Stokes/convection-diffusion, is discretized using HDG, while the solid part, governed by heat equation, is discretized using CG. In the end, the results of coupled CG-CG model, where all the equations are discretized by CG, are also presented for important quantities of interest like Nusselt number.

The geometry of the problem along with the applied boundary conditions are shown in fig. 3.16. The fluid domain, Ω_1 , is the square cavity of unit length and the solid wall, Ω_2 , has a thickness of 0.2. The ratio of thermal diffusivities of solid to fluid is considered as unity. Prandtl number (Pr) of 0.71 is used in Ω_1 in the computations.

The problem is solved with the proposed $\text{CG}(Q_{k+1})\text{-HDG}(Q_k)$ model and, also, a $\text{CG}(Q_{k+1})\text{-CG}(Q_{k+1}Q_k)$ model. The solution is approximated with one degree higher in the CG domain than HDG, to keep the optimal convergence in both domains, as discussed in Section 3.3. The obtained results are compared with the literature data. A degree of approximation $k = 5$, and an uniform mesh size, $h = 0.025$, with quadrilateral elements are used in both models. As stated in the Section 3.2.4, at higher Ra number, an incremental method is used to obtain the steady state numerical solution. The stabilization parameters are the same as in table 3.1.

Figure 3.17 presents the isolines for velocity and temperature at different Ra number. The temperature and velocity distributions inside the fluid cavity are similar to the solutions of Rayleigh–Bénard convection flow with relatively less sharp boundary layer. At higher Ra , the

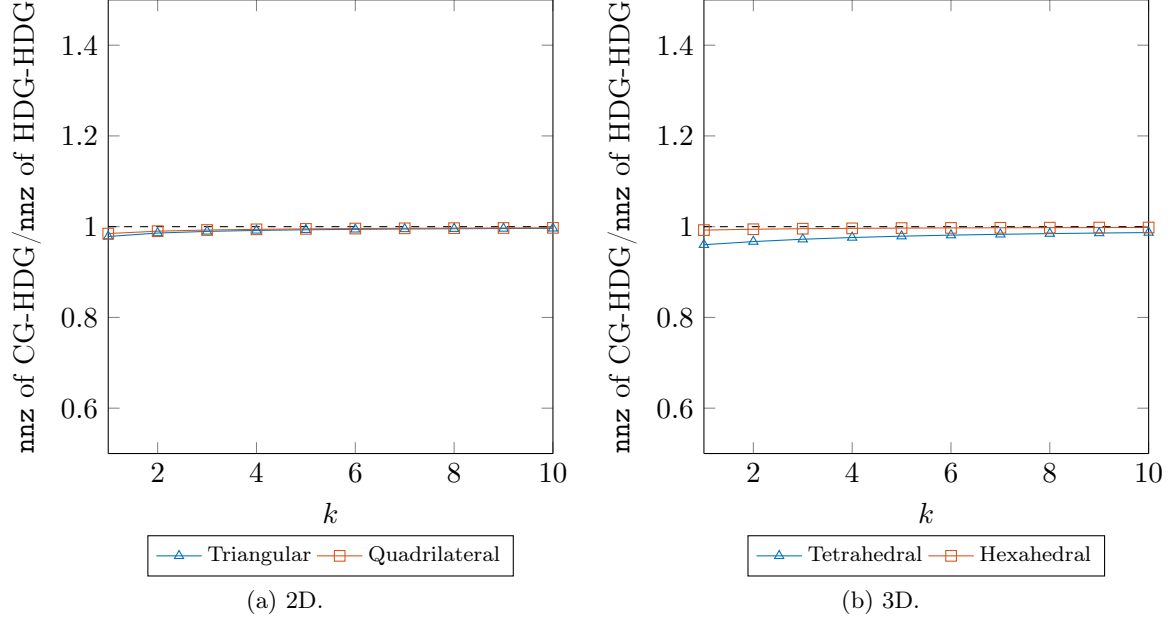


Figure 3.15: Comparison of nnz between coupled $\text{CG}(T_{k+1})\text{-HDG}(T_k)$ and $\text{HDG}(T_k)\text{-HDG}(T_k)$ models.

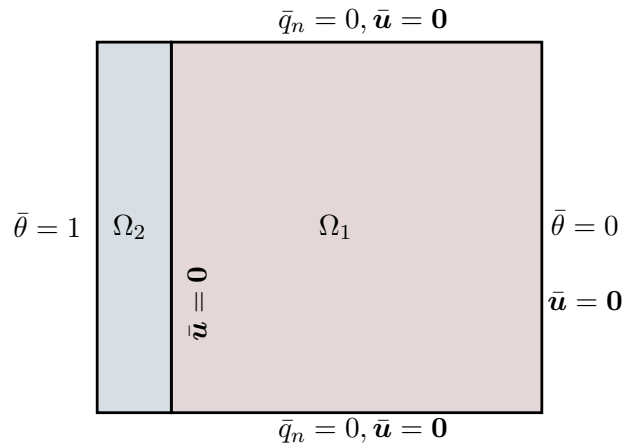


Figure 3.16: Conjugate heat transfer problem: geometry and prescribed boundary values of conjugate heat transfer problem domain.

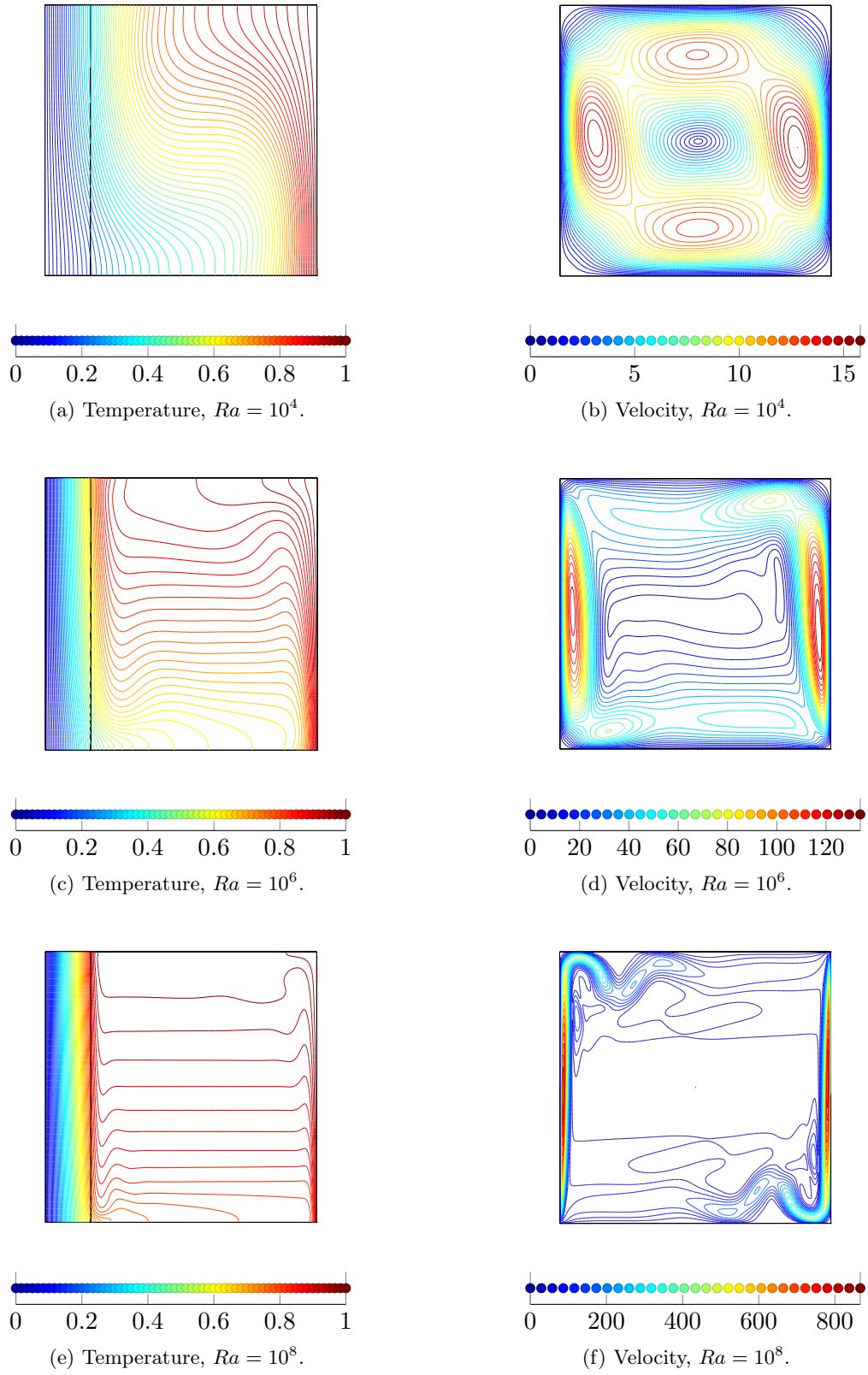


Figure 3.17: Conjugate heat transfer problem: isolines of temperature and velocity at different Ra numbers using CG-HDG model. Post-processed temperature and velocity are presented in Ω_1 .

Table 3.6: Conjugate heat transfer problem: summary of important quantities and comparison with literature values for $Ra = 10^3 - 10^8$.

Ra	Quantity	Misra & Sarkar 1997	Kaminski & Prakash 1986	CG(Q_6)-HDG(Q_5) / CG(Q_6)-CG(Q_6Q_5)
0.7×10^3	\overline{Nu}	—	0.87	0.8678
	Mean θ along interface	—	—	0.8264
10^3	\overline{Nu}	0.8958	—	0.8957
	Mean θ along interface	0.1791	—	0.1791
10^4	\overline{Nu}	1.4528	—	1.4537
	Mean θ along interface	0.2906	—	0.2906
0.7×10^5	\overline{Nu}	—	2.08	2.0850
	Mean θ along interface	—	—	0.5832
10^5	\overline{Nu}	2.1997	—	2.2014
	Mean θ along interface	0.4400	—	0.4400
0.7×10^6	\overline{Nu}	—	2.87	2.8540
	Mean θ along interface	—	—	0.4294
10^6	\overline{Nu}	2.9528	—	2.9605
	Mean θ along interface	0.5916	—	0.5919
0.7×10^7	\overline{Nu}	—	3.53	3.5077
	Mean θ along interface	—	—	0.2987
10^7	\overline{Nu}	—	—	3.5913
	Mean θ along interface	—	—	0.7181
10^8	\overline{Nu}	—	—	4.061
	Mean θ along interface	—	—	0.8120

loss of symmetry in the solution of velocity can be clearly noticed in the present example.

Average Nusselt number at the fluid-solid interface is computed and compared with the values present in the literature. Whenever possible, the average temperature on the interface is also compared with literature data. It is to be noted that the geometry considered in the work of Kaminski and Prakash (1986) has the solid wall to the right of the fluid domain. Hence, the results presented for the comparison with the mentioned work correspond to the solid wall to the right of the fluid domain. All the results obtained by both models are presented in table 3.6 and, for the number of significant digits presented, they coincide. They are also very close to the literature values.

The models will be now compared in terms of computational effort to point their relative merits. The computational details of CG(Q_{k+1})-HDG(Q_k) and CG(Q_{k+1})-CG($Q_{k+1}Q_k$) models are tabulated in table 3.7, similar to one presented for Rayleigh–Bénard convection flow. It is clear from this table that the proposed CG-HDG model is superior to CG-CG model in terms of computational efficiency and complexity. It is also worth stressing that both models have similar orders of convergence for all the variables of interest. Hence, the proposed CG-HDG model can be beneficial to use for multi-physics flows involving high Ra number.

Table 3.7: Conjugate heat transfer problem: computational details for CG-HDG and CG-CG for $Ra = 10^5$ and $h = 0.025$.

	CG(Q_6)-HDG(Q_5)	CG(Q_6)-CG(Q_6Q_5)
No. of DOFs, ndof	64 209	72 612
CPU time for linear solver	13.3 sec	25.5 sec
No. of non-linear iterations	11	11
Condition number, $\kappa(\mathbf{A})$	$\mathcal{O}(10^{10})$	$\mathcal{O}(10^{12})$
No. of non-zeros, nnz	7 377 875	11 033 375

3.5 Conclusions

Optimal convergence and super-convergence rates are shown in the numerical tests for the coupled Navier–Stokes/convection-diffusion equations formulation with HDG. A study on computational efficiency is also presented for this problem and concluded that for the same level of accuracy HDG takes lesser CPU time for linear solver than CG when degree of approximation is greater than 2, which is the same conclusion drawn from study in Chapter 2.

The results of Rayleigh–Bénard convection flow are presented until Ra of 10^8 and the Nusselt numbers are compared between HDG, CG and literature values. Even though, both HDG and CG can resolve the solution at high Ra and show similar level of accuracy, it is noticed that HDG has less computational cost. CPU times for linear solver and condition numbers of HDG are lower than its CG counterpart for the same mesh and degree of approximation in the example considered.

A coupled HDG and CG formulation for the heat equation is proposed and tested with numerical examples. As expected, optimal HDG and CG convergence rates are kept with the proposed CG-HDG coupled formulation for the heat equation when the degree of approximation for CG is one degree higher than HDG degree.

Finally, the proposed CG-HDG coupling is merged with coupled HDG formulation for Navier–Stokes/convection-diffusion equations and optimal convergence rates are established. The relative CPU times for linear solver is compared between CG-HDG and CG-CG models and concluded CG-HDG model is more efficient. A similar study is performed between CG-HDG and HDG-HDG models and inferred that both the models are very close in terms of CPU times for linear solver. The theoretical count on number of non-zero entries in CG-HDG and HDG-HDG models are also very similar. Hence, both CG-HDG and HDG-HDG models showed similar performance and memory requirements for considered examples.

The benchmark problem of conjugate heat transfer is solved using CG-HDG and CG-CG models and compared to literature data. Both models give the identical results in terms of Nusselt numbers and mean temperature on interface, but, it is shown that CG-HDG model has higher computational efficiency than CG-CG model.

Chapter 4

Application to Glass Fiber Reinforced Polymer tubular cross-section

4.1 Overview

This chapter presents the results of thermal response of a GFRP tubular cross-section subjected to fire. The problem description and corresponding governing equations are provided. The radiosity equation that governs the internal radiation is discussed in brief along its discretization details. An overview of the computation of discretization error and numerical uncertainty is presented in the context of practical application problems where analytical solution is not available. Finally, various numerical results of GFRP cross-section are presented and validated with the available experimental data.

4.2 Problem description

4.2.1 Geometry of GFRP cross-section

The geometry of the GFRP tubular cross-section considered in the analysis is presented in fig. 4.1. The length of the cross-section is 0.1 m with a GFRP thickness of 0.008 m. The specimen used in the experiments has rounded corners in the internal cavity with a fillet radius of 0.003 m. Most of the analysis is made with the cross-section with straight corners as shown in the fig. 4.1. This simplification in the model allows to generate structured meshes. Nevertheless, results are also provided for the geometry with rounded corners.

4.2.2 Governing equations

Let Ω_s represent the GFRP material and Ω_f represent the air cavity inside GFRP tubular cross-section. The interface between Ω_s and Ω_f is designated as Γ_R , *i.e.*, $\Gamma_R = \partial\Omega_s \cap \partial\Omega_f$. The exterior boundaries of the Ω_s are divided into Γ_t , Γ_b , Γ_l and Γ_r . The governing equations in both Ω_s and Ω_f along with the boundary conditions are given as follows,

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \operatorname{div}(\mathbf{u} \otimes \mathbf{u}) - \operatorname{div}(-p\mathbf{I} + \nu \operatorname{grad} \mathbf{u}) - \mathbf{f}(\theta_f) &= \mathbf{0} \quad \text{in } \Omega_f, \\ \operatorname{div} \mathbf{u} &= 0 \quad \text{in } \Omega_f, \\ \frac{\partial \theta_f}{\partial t} + \operatorname{div}(-\alpha_f \operatorname{grad} \theta_f + \mathbf{u} \theta_f) &= 0 \quad \text{in } \Omega_f, \end{aligned} \tag{4.1a}$$

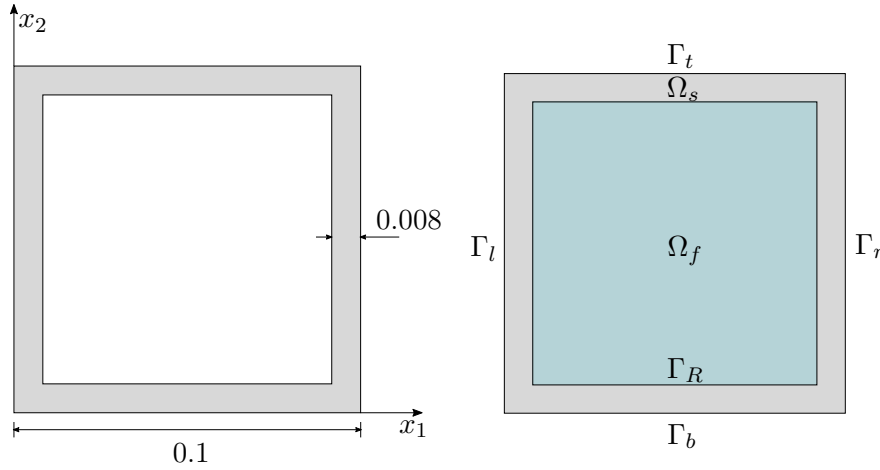


Figure 4.1: Geometry and boundary conditions notations used for GFRP tubular cross-section. All dimensions are in meters.

$$\frac{\partial \theta_s}{\partial t} + \operatorname{div}(-\alpha_s(\theta_s) \operatorname{grad} \theta_s) = 0 \quad \text{in } \Omega_s, \quad (4.1b)$$

$$\theta_f - \theta_s = 0 \quad \text{on } \Gamma_R, \quad (4.1c)$$

$$-(\kappa_f \operatorname{grad} \theta_f) \cdot \mathbf{n}_f - (\kappa_s(\theta_s) \operatorname{grad} \theta_s) \cdot \mathbf{n}_s - \frac{\epsilon}{1 - \epsilon} (\sigma \theta_s^4 - R) = 0 \quad \text{on } \Gamma_R, \quad (4.1d)$$

$$\mathbf{u} = \mathbf{0} \quad \text{on } \Gamma_R, \quad (4.1e)$$

$$-(\kappa_s(\theta_s) \operatorname{grad} \theta_s) \cdot \mathbf{n} + h_a(\theta_s) (\theta_a - \theta_s) + \epsilon \sigma (\theta_s^4 - \theta_a^4) = 0 \quad \text{on } \Gamma_t \cup \Gamma_b, \quad (4.1f)$$

$$-(\kappa_s(\theta_s) \operatorname{grad} \theta_s) \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_l \cup \Gamma_r, \quad (4.1g)$$

with initial conditions $\theta_s = \theta_f = \theta_0$ and $\mathbf{u} = \mathbf{0}$ at $t = 0$. Equations (4.1a) are the transient incompressible Navier–Stokes equations coupled with convection-diffusion equation, where \mathbf{u} is velocity of fluid, p is kinematic pressure, θ_f is the temperature of the air, ν is the viscosity and α_f is the thermal diffusivity of the air $\left(\frac{\kappa_f}{\rho_f c_p^f} \right)$. The natural buoyancy of the air is modelled using the Boussinesq approximation and the term $\mathbf{f}(\theta_f)$ is defined as,

$$\mathbf{f}(\theta_f) = -\beta \mathbf{g}(\theta_f - \theta_0), \quad (4.2)$$

where β is the Boussinesq term, \mathbf{g} is the gravity force and θ_0 is the reference temperature. Equation (4.1b) is the heat conduction equation governing the temperature distribution in solid domain, Ω_s , where, θ_s is the temperature of the GFRP and α_s is the thermal diffusivity of the GFRP, which depends on temperature. The transmission conditions on the interface of fluid and solid domains are presented in equations (4.1c) and (4.1d) which represents continuity of temperature and equilibrium of normal component of the fluxes, respectively. In equation (4.1d), R is the radiosity and it is discussed in the next section.

No-slip boundary condition is prescribed on all the boundary of the fluid. In the case of the exterior boundary of the solid, a combination of convective and radiative boundary condition is applied on Γ_t and Γ_b in equation (4.1f), while homogeneous Neumann boundary conditions are applied on Γ_l and Γ_r in equation (4.1g). In equation (4.1f), h_a stands for convective heat transfer coefficient, σ is the Stefan–Boltzmann constant, ϵ is the emissivity and θ_a is defined

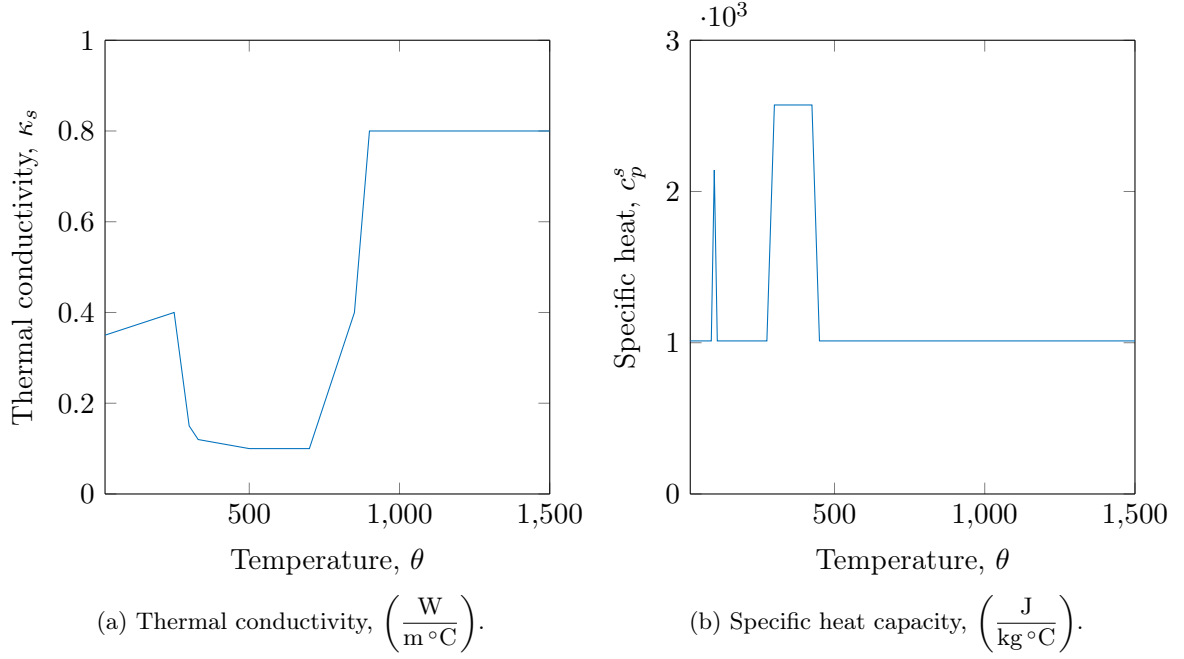


Figure 4.2: Thermal properties of GFRP material as a function of temperature.

as follows,

$$\theta_a = \begin{cases} \theta_0 & \text{on } \Gamma_t, \\ \theta_0 + 345 \log(8t + 1) & \text{on } \Gamma_b, \end{cases} \quad (4.3)$$

where t is the time (in minutes). Equation (4.3) represents that the top face of GFRP cross-section is at the reference (ambient) temperature, while the bottom face is heated according to temperature profile given in ISO 834 (1975).

4.2.3 Material properties

As stated earlier, the thermo-physical properties of GFRP are temperature-dependent. In the present analysis, the material properties proposed by Tracy (2005) are used, as suggested in the work of López (2017). Figure 4.2 shows the dependence of thermal conductivity and specific heat capacity of GFRP material with the temperature. The variation of density of the GFRP with temperature is shown in fig. 4.3. The emissivity, ϵ , of the GFRP is assumed as constant value of 0.75.

The thermal properties of air are assumed to be constant throughout the analysis. The viscosity, ν , is taken as $1.67 \times 10^{-5} \text{ m}^2/\text{s}$, thermal conductivity, κ_f , is $0.0256912 \left(\frac{\text{W}}{\text{m}^\circ\text{C}}\right)$, specific heat capacity, c_p^f , is $1004.592 \left(\frac{\text{J}}{\text{kg}^\circ\text{C}}\right)$ and density, ρ^f , is $1.2 \text{ kg}/\text{m}^3$. The Boussinesq parameter, β , is taken as $0.003 \text{ }^\circ\text{C}^{-1}$ and vertical component of the gravity force vector is set to $-9.81 \text{ m}/\text{s}^2$. The convective heat transfer coefficient of air, h_a , is also temperature dependent and it is computed by linear interpolation between 20°C and 1000°C where h_a varies between 5 and 50 $\left(\frac{\text{W}}{\text{m}^2^\circ\text{C}}\right)$. The Stefan–Boltzmann constant, σ , is set to $5.669 \times 10^{-8} \left(\frac{\text{W}}{\text{m}^2^\circ\text{C}^4}\right)$.

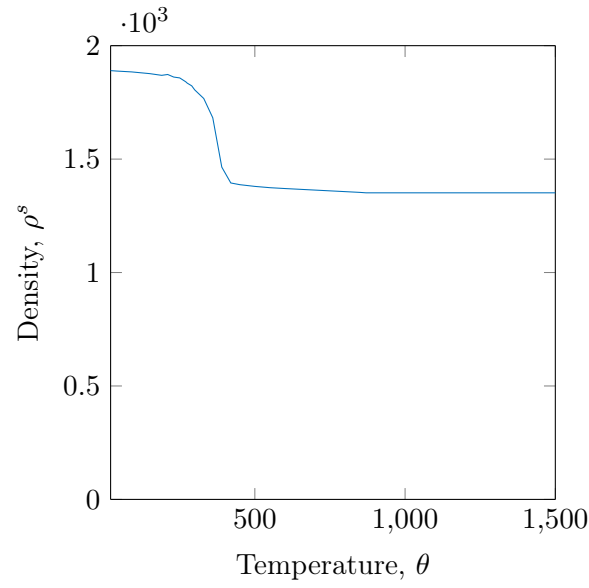


Figure 4.3: Density of GFRP material as a function of temperature.

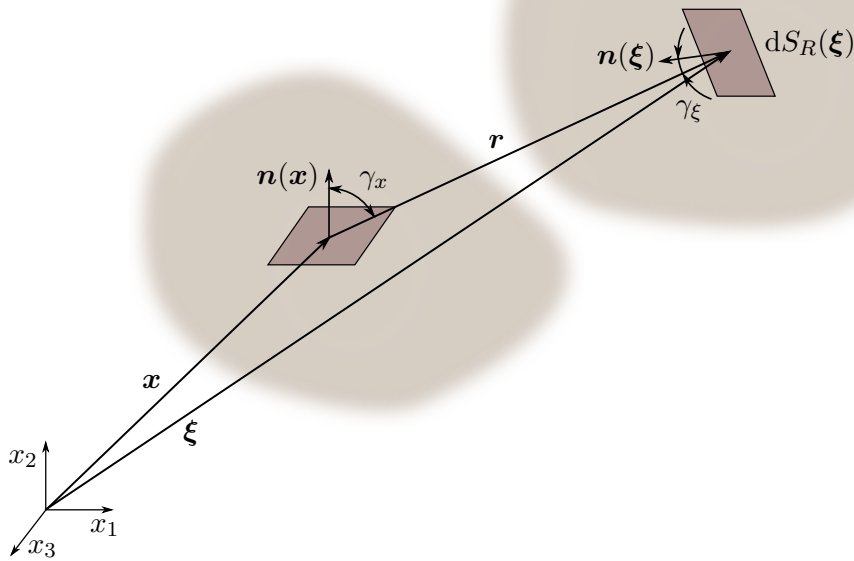


Figure 4.4: Schematic representation of radiative exchange between surfaces of a cavity.

4.3 Radiosity

4.3.1 Analytical equation

The governing equations presented in (4.1) are not closed until a suitable definition for radiosity, R , is established. Figure 4.4 shows the schematic representation of the radiative exchange in a cavity between two generic surfaces in 3-D space. This radiative flux is computed using

the following equation,

$$R = \sigma \epsilon \theta^4 + (1 - \epsilon) \int_{S_R} R \frac{\cos \gamma_x \cos \gamma_\xi}{\pi r^2} dS_R, \quad (4.4)$$

where the angles γ_x and γ_ξ are the angles made by the vector, $\mathbf{r} = \boldsymbol{\xi} - \mathbf{x}$ with the respective normals to the two surfaces and r is the norm of the vector, \mathbf{r} . The particularization of equation (4.4) for a two-dimensional case can be expressed as follows (López *et al.*, 2014),

$$R = \sigma \epsilon \theta^4 + (1 - \epsilon) \int_{\Gamma_R} R \frac{\cos \gamma_x \cos \gamma_\xi}{2r} d\Gamma_R, \quad (4.5)$$

where Γ_R is a generic boundary of a surface.

4.3.2 Discretization

Radiosity equation is an algebraic equation that is imposed in weak form. Using δR , which is arbitrary weighing function, the weak form of the radiosity equation can be expressed as follows,

$$\left\langle \delta R, \frac{R}{1 - \epsilon} \right\rangle_{\Gamma_R} = \left\langle \delta R, \sigma \frac{\epsilon}{1 - \epsilon} \theta^4 \right\rangle_{\Gamma_R} + \left\langle \delta R, \left\langle R, \frac{\cos \gamma_x \cos \gamma_\xi}{2r} \right\rangle_{\Gamma_R} \right\rangle_{\Gamma_R}. \quad (4.6)$$

Equation (4.6) is obtained by multiplying (4.5) with δR and dividing by $1 - \epsilon$ throughout. The internal radiative boundary can be discretized into several line elements along the boundary. Each line element interacts with all the other elements in absence of obstacles between them, *i.e.* the straight line joining two elements should not intersect the boundary. If e and k are two generic line elements, the weak form for radiosity can be rewritten as

$$\left\langle \delta R^e, \frac{R^e}{(1 - \epsilon)} \right\rangle_{\Gamma_R^e} = \left\langle \delta R^e, \sigma \frac{\epsilon}{(1 - \epsilon)} (\theta^e)^4 \right\rangle_{\Gamma_R^e} + \sum_{\substack{k=1 \\ k \neq e}}^{n_l} \left\langle \delta R^e, \left\langle R^k, \frac{\cos \gamma^e \cos \gamma^k}{2r} \right\rangle_{\Gamma_R^k} \right\rangle_{\Gamma_R^e}, \quad (4.7)$$

where γ^e and γ^k are the angles made by the line joining the elements e and k with their respective internal normals. Here, n_l is the number of elements on the interface, Γ_R .

4.4 Coupled CG-HDG formulation

4.4.1 Weak formulation

While presenting the weak formulation for the considered problem, the notation introduced in the previous chapters is re-defined for the mesh skeleton. The mesh skeleton, Γ_f , is defined as,

$$\Gamma_f = \bigcup_{e=1}^{m_{el}} \partial \Omega_f^e, \quad (4.8)$$

where m_{el} is the number of elements in Ω_f . Similarly, p_{el} is the number of elements in Ω_s .

The CG-HDG coupled weak formulation with BDF time integration scheme for the governing equations presented in (4.1) and (4.5) can be stated as: find $(\mathbf{u}^{n+1}, p^{n+1}, \mathbf{L}^{n+1}, \theta_f^{n+1}, \mathbf{q}_f^{n+1}, \hat{\mathbf{u}}^{n+1}, \rho_e^{n+1}, \hat{\theta}_f^{n+1}, \theta_s^{n+1}, R^{n+1}) \in [\mathcal{V}_k^h(\Omega_f)]^2 \times \mathcal{V}_k^h(\Omega_f) \times [\mathcal{V}_k^h(\Omega_f)]^{2 \times 2} \times \mathcal{V}_k^h(\Omega_f) \times [\mathcal{V}_k^h(\Omega_f)]^2 \times$

$[\Lambda_k^h(\Gamma_f \setminus \Gamma_R)]^2 \times \mathbb{R}^{m_{el}} \times \Lambda_k^h(\Gamma_f \setminus \Gamma_R) \times \mathcal{W}_{k+1}^h(\Omega_s) \times \Sigma_{k+1}^h(\Gamma_R)$ such that $\mathbf{u}^0 = \mathbf{0}$, $\theta_s^0 = \theta_f^0 = \theta_0$ and

$$\begin{aligned} & (\delta \mathbf{L}, \mathbf{L}^{n+1})_{\Omega_f^e} + (\operatorname{div} \delta \mathbf{L}, \mathbf{u}^{n+1})_{\Omega_f^e} - \langle \delta \mathbf{L} \mathbf{n}, \hat{\mathbf{u}}^{n+1} \rangle_{\partial \Omega_f^e} = 0, \\ & \left(\delta \mathbf{u}, \frac{\alpha_\mu \mathbf{u}^{n+1}}{\Delta t^{n+1}} \right)_{\Omega_f^e} - (\operatorname{grad} \delta \mathbf{u}, \mathbf{u}^{n+1} \otimes \mathbf{u}^{n+1})_{\Omega_f^e} - (\delta \mathbf{u}, \operatorname{div} (-p^{n+1} \mathbf{I} + \nu \mathbf{L}^{n+1}))_{\Omega_f^e} \\ & \quad + \langle \delta \mathbf{u}, (\hat{\mathbf{u}}^{n+1} \otimes \hat{\mathbf{u}}^{n+1}) \mathbf{n} + \tau_u (\mathbf{u}^{n+1} - \hat{\mathbf{u}}^{n+1}) \rangle_{\partial \Omega_f^e} \\ & \quad - \left(\delta \mathbf{u}, \mathbf{f}(\theta_f^{n+1}) \right)_{\Omega_f^e} - \left(\delta \mathbf{u}, \frac{\mathbf{u}^{n, \text{BDF}\mu}}{\Delta t^{n+1}} \right)_{\Omega_f^e} = 0, \end{aligned} \quad (4.9a)$$

$$\begin{aligned} & - (\operatorname{grad} \delta p, \mathbf{u}^{n+1})_{\Omega_f^e} + \langle \delta p, \hat{\mathbf{u}}^{n+1} \cdot \mathbf{n} \rangle_{\partial \Omega_f^e} = 0, \\ & \frac{1}{|\partial \Omega_f^e|} \langle p^{n+1}, 1 \rangle_{\partial \Omega_f^e} = \rho_e^{n+1}, \end{aligned} \quad (4.9b)$$

$$\begin{aligned} & \left(\delta \theta_f, \rho_f c_{p_f} \frac{\alpha_\mu \theta_f^{n+1}}{\Delta t^{n+1}} \right)_{\Omega_f^e} + \left(\delta \theta_f, \rho_f c_{p_f} \operatorname{div} \mathbf{q}_f^{n+1} \right)_{\Omega_f^e} - \left(\operatorname{grad} \delta \theta_f, \rho_f c_{p_f} \mathbf{u}^{n+1} \theta_f^{n+1} \right)_{\Omega_f^e} \\ & \quad + \langle \delta \theta_f, \rho_f c_{p_f} (\hat{\mathbf{u}}^{n+1} \cdot \mathbf{n}) \hat{\theta}_f^{n+1} \rangle_{\partial \Omega_f^e \setminus \Gamma_R} + \left\langle \delta \theta_f, \rho_f c_{p_f} \tau_\theta (\theta_f^{n+1} - \hat{\theta}_f^{n+1}) \right\rangle_{\partial \Omega_f^e \setminus \Gamma_R} \\ & \quad + \left\langle \delta \theta_f, \rho_f c_{p_f} \tau_\theta (\theta_f^{n+1} - \mathbb{P}_2(\theta_s^{n+1})) \right\rangle_{\partial \Omega_f^e \cap \Gamma_R} - \left(\delta \theta_f, \rho_f c_{p_f} \frac{\theta_f^{n, \text{BDF}\mu}}{\Delta t^{n+1}} \right)_{\Omega_f^e} = 0, \end{aligned} \quad (4.9c)$$

$$\begin{aligned} & \left(\delta \mathbf{q}_f, \alpha_f^{-1} \mathbf{q}_f^{n+1} \right)_{\Omega_f^e} - \left(\operatorname{div} \delta \mathbf{q}_f, \theta_f^{n+1} \right)_{\Omega_f^e} + \left\langle \delta \mathbf{q}_f \cdot \mathbf{n}, \hat{\theta}_f^{n+1} \right\rangle_{\partial \Omega_f^e \setminus \Gamma_R} \\ & \quad + \left\langle \delta \mathbf{q}_f \cdot \mathbf{n}_f, \mathbb{P}_2(\theta_s^{n+1}) \right\rangle_{\partial \Omega_f^e \cap \Gamma_R} = 0, \end{aligned} \quad (4.9d)$$

for $e = 1, \dots, m_{el}$, and,

$$\begin{aligned} & \sum_{e=1}^{m_{el}} \langle \delta \hat{\mathbf{u}}, (-p^{n+1} \mathbf{I} + \nu \mathbf{L}^{n+1}) \mathbf{n} + \tau_u (\hat{\mathbf{u}}^{n+1} - \mathbf{u}^{n+1}) \rangle_{\partial \Omega_f^e} = 0, \\ & \langle \hat{\mathbf{u}}^{n+1} \cdot \mathbf{n}, 1 \rangle_{\partial \Omega_f^e} = 0 \quad \text{for } e = 1, \dots, m_{el}, \end{aligned} \quad (4.10a)$$

$$\hat{\mathbf{u}}^{n+1} = \mathbf{0} \quad \text{on } \Gamma_R,$$

$$\sum_{e=1}^{m_{el}} \left\langle \delta \hat{\theta}_f, \left(\rho_f c_{p_f} (\mathbf{q}_f^{n+1} + \hat{\mathbf{u}}^{n+1} \hat{\theta}_f^{n+1}) \cdot \mathbf{n} + \rho_f c_{p_f} \tau_\theta (\theta_f^{n+1} - \hat{\theta}_f^{n+1}) \right) \right\rangle_{\partial \Omega_f^e \setminus \Gamma_R} = 0, \quad (4.10b)$$

$$\begin{aligned} & \left(\delta \theta_s, \rho_s (\theta_s^{n+1}) c_{p_s} (\theta_s^{n+1}) \frac{\alpha_\mu \theta_s^{n+1}}{\Delta t^{n+1}} \right)_{\Omega_s} + (\operatorname{grad} \delta \theta_s, \kappa_s (\theta_s^{n+1}) \operatorname{grad} \theta_s^{n+1})_{\Omega_s} \\ & \quad - \left\langle \mathbb{P}_2(\delta \theta_s), \rho_f c_{p_f} \mathbf{q}_f^{n+1} \cdot \mathbf{n}_f + \tau_\theta (\theta_f^{n+1} - \mathbb{P}_2(\theta_s^{n+1})) - \frac{\epsilon}{1-\epsilon} (\sigma (\theta_s^{n+1})^4 - R^{n+1}) \right\rangle_{\Gamma_R} \\ & \quad - \left(\delta \theta_s, \rho_s (\theta_s^{n+1}) c_{p_s} (\theta_s^{n+1}) \frac{\theta_s^{n, \text{BDF}\mu}}{\Delta t^{n+1}} \right)_{\Omega_s} - \langle \delta \theta_s, h_a (\theta_s^{n+1}) (\theta_a^{n+1} - \theta_s^{n+1}) \rangle_{\Gamma_t \cup \Gamma_b} \\ & \quad - \langle \delta \theta_s, \epsilon \sigma ((\theta_a^{n+1})^4 - (\theta_s^{n+1})^4) \rangle_{\Gamma_t \cup \Gamma_b} = 0, \end{aligned} \quad (4.10c)$$

$$\begin{aligned}
\sum_{e=1}^{n_l} \left\langle \delta R, \frac{R^{n+1}}{(1-\epsilon)} \right\rangle_{\Gamma_R^e} - \sum_{e=1}^{n_l} \left\langle \delta R, \sigma \frac{\epsilon}{(1-\epsilon)} (\theta_s^{n+1})^4 \right\rangle_{\Gamma_R^e} \\
- \sum_{e=1}^{n_l} \sum_{\substack{k=1 \\ k \neq e}}^{n_l} \left\langle \delta R, \left\langle R^{n+1,k}, \frac{\cos \gamma^e \cos \gamma^k}{2r} \right\rangle_{\Gamma_R^k} \right\rangle_{\Gamma_R^e} = 0,
\end{aligned} \tag{4.10d}$$

for all $(\delta \mathbf{u}, \delta p, \delta \mathbf{L}, \delta \theta_f, \delta \mathbf{q}_f, \delta \hat{\mathbf{u}}, \delta \theta_f, \delta \theta_s, \delta R) \in [\mathcal{V}_k^h(\Omega_f)]^2 \times \mathcal{V}_k^h(\Omega_f) \times [\mathcal{V}_k^h(\Omega_f)]^{2 \times 2} \times \mathcal{V}_k^h(\Omega_f) \times [\mathcal{V}_k^h(\Omega_f)]^2 \times [\Lambda_k^h(\Gamma_f \setminus \Gamma_R)]^2 \times \Lambda_k^h(\Gamma_f \setminus \Gamma_R) \times \mathcal{W}_{k+1}^h(\Omega_s) \times \Sigma_{k+1}^h(\Gamma_R)$ such that $\delta \hat{\mathbf{u}} = \mathbf{0}$ on Γ_R where the discrete spaces are defined in (1.2). Note that the term $\langle (\mathbb{P}_2(\delta \theta_s) - \delta \theta_s), \alpha_s \text{grad } \theta_s \cdot \mathbf{n}_s \rangle_{\Gamma_R}$ is already neglected while presenting the weak form. Equations (4.9c) and (4.10b) are multiplied by $\rho_f c_{pf}$ to be dimensionally consistent with the transmission conditions in (4.1d). The continuity in flux condition (4.1d) is stated using thermal conductivity coefficients, κ_s and κ_f , unlike thermal diffusivity coefficients, α_s and α_f , in the previous examples presented in this work.

4.4.2 Residual in matrix notation

The matrix form of the residual vector corresponding to weak forms (4.9) and (4.10) at the time step $n + 1$ can be expressed as follows,

$$\begin{bmatrix} \mathbf{K}_{gg} & \mathbf{K}_{gl} \\ \mathbf{K}_{lg} & \mathbf{K}_{ll} \end{bmatrix} \begin{Bmatrix} \mathbf{x}_g \\ \mathbf{x}_l \end{Bmatrix} - \begin{Bmatrix} \mathbf{f}_g \\ \mathbf{f}_l \end{Bmatrix} = \mathbf{0}, \tag{4.11}$$

where the suffixes g and l represent global and local variables, respectively and components in the residual are given in (4.12). The elemental matrices that arise from radiosity equation are provided in Appendix C. The tangent operators of the non-linear contributions are already provided in (3.45) and the only new terms in the present case are the non-linear contributions from the radiosity equation. They are also presented in Appendix C. The dependence of ρ_s and c_p^s on temperature is omitted in equation (4.12) for the ease of representation. Nevertheless, the tangent operators are computed taking the non-linear dependence of materials properties into consideration.

4.4.3 Methodology

As presented in the case of conjugate heat transfer example, the present problem is solved using CG-HDG and CG-CG models. Both quadrilateral and triangular elements are considered in the analysis of the problem. Radiosity equation is discretized using 1-D high-order line elements based on Gauss–Lobatto points. The degree of approximation on radiosity elements is always the same as degree in Ω_s , in both CG-HDG and CG-CG models.

The time dependent terms are discretized using the BDF schemes presented earlier for Navier–Stokes equations. In the present example, results from both constant time step and adaptive time step are presented. The adaptive time stepping scheme presented in Algorithm 1 in Section 2.8.4 is used. The extension of the algorithm presented for Navier–Stokes to the present conjugate heat transfer problem is straightforward. Instead of using the estimated local error from velocity alone, local error from temperature in Ω_f and Ω_s are also considered in adapting the time step. The limiting constants used in this problem are the same ones that are provided in Section 2.8.4.

$$\begin{aligned}
\mathbf{K}_{gg} &= \begin{bmatrix} \mathbf{A}_{\hat{u}\hat{u}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{\hat{\rho}\hat{u}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \rho_f c_p^f \mathbf{A}_{\hat{\theta}\hat{\theta}}(\hat{\mathbf{u}}) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \rho_f c_p^f \mathbf{P}^T \mathbf{A}_{\hat{\theta}\hat{\theta}}(\hat{\mathbf{u}}) \mathbf{P} & \rho_s c_p^s \mathbf{K}_{\theta\theta}^{i_i} & \rho_s c_p^s \mathbf{K}_{\theta\theta}^{i_i} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & + \rho_s c_p^s \mathbf{K}_{\theta\theta}^{II} + \frac{\rho_s c_p^s}{\Delta t} \overline{\mathbf{M}}_{\theta\theta}^{II} & & + \frac{\rho_s c_p^s}{\Delta t} \overline{\mathbf{M}}_{\theta\theta}^{I_i} & & & & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \rho_s c_p^s \mathbf{K}_{p\theta\theta}^{iI} & \rho_s c_p^s \mathbf{K}_{p\theta\theta}^{iI} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & + \frac{\rho_s c_p^s}{\Delta t} \overline{\mathbf{M}}_{\theta\theta}^{iI} & & + \frac{\rho_s c_p^s}{\Delta t} \overline{\mathbf{M}}_{\theta\theta}^{iI} & & & & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{K}_{RR} & \mathbf{0} \end{bmatrix}^{n+1}, \mathbf{K}_{gl} = \begin{bmatrix} \mathbf{A}_{\hat{u}u} & \mathbf{A}_{\hat{u}L} & \mathbf{A}_{\hat{u}p} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \rho_f c_p^f \mathbf{A}_{\hat{\theta}\theta} & \rho_f c_p^f \mathbf{A}_{\hat{\theta}q} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \rho_f c_p^f \mathbf{P}^T \mathbf{A}_{\hat{\theta}\theta} & \rho_f c_p^f \mathbf{P}^T \mathbf{A}_{\hat{\theta}q} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}^{n+1}, \\
\mathbf{K}_{lg} &= \begin{bmatrix} \mathbf{A}_{u\hat{u}}(\hat{\mathbf{u}}) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{L\hat{u}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{p\hat{u}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \rho_f c_p^f \mathbf{A}_{\theta\hat{\theta}}(\hat{\mathbf{u}}) & \rho_f c_p^f \mathbf{A}_{\theta\hat{\theta}}(\hat{\mathbf{u}}) \mathbf{P} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{q\hat{\theta}} & \mathbf{A}_{q\theta} \mathbf{P} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}^{n+1}, \mathbf{K}_{ll} = \begin{bmatrix} \mathbf{A}_{uu}(\mathbf{u}) & \mathbf{A}_{uL} & \mathbf{A}_{up} & \mathbf{0} & \mathbf{A}_{u\theta} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ + \frac{1}{\Delta t} \mathbf{M}_{uu} & & & & & & & \\ \mathbf{A}_{Lu} & \mathbf{A}_{LL} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{pu} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{pp}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{pp} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \rho_f c_p^f \mathbf{A}_{\theta\theta}(\mathbf{u}) & \rho_f c_p^f \mathbf{A}_{\theta q} & \mathbf{0} & \mathbf{0} \\ & & & & + \frac{\rho_f c_p^f}{\Delta t} \mathbf{M}_{\theta\theta} & & & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{q\theta} & \mathbf{A}_{qq} & \mathbf{0} & \mathbf{0} \end{bmatrix}^{n+1}, \\
\mathbf{x}_g &= \left\{ \hat{\mathbf{u}}^{n+1} \quad \boldsymbol{\rho}^{n+1} \quad \hat{\boldsymbol{\theta}}_f^{n+1} \quad \boldsymbol{\theta}_s^{I,n+1} \quad \boldsymbol{\theta}_s^{i,n+1} \quad \mathbf{R}^{n+1} \right\}^T, \\
\mathbf{x}_l &= \left\{ \mathbf{u}^{n+1} \quad \mathbf{L}^{n+1} \quad \mathbf{p}^{n+1} \quad \boldsymbol{\lambda}^{n+1} \quad \boldsymbol{\theta}_f^{n+1} \quad \mathbf{q}_f^{n+1} \right\}^T, \\
\mathbf{f}_g &= \left\{ \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{f}_\theta^{I,n+1}(\boldsymbol{\theta}_s, \mathbf{R}) + \frac{\rho_s c_p^s}{\Delta t} \overline{\mathbf{M}}_{\theta\theta} \boldsymbol{\theta}_s^{I,n+1} \quad \mathbf{f}_\theta^{i,n+1}(\boldsymbol{\theta}_s, \mathbf{R}) + \frac{\rho_s c_p^s}{\Delta t} \overline{\mathbf{M}}_{\theta\theta} \boldsymbol{\theta}_s^{i,n+1} \quad \mathbf{f}_R^{n+1} \right\}^T, \\
\mathbf{f}_l &= \left\{ \tilde{\mathbf{f}}_{\theta_0}^{n+1} + \frac{1}{\Delta t} \mathbf{M}_{uu} \mathbf{u}^{n+1} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \frac{\rho_f c_p^f}{\Delta t} \mathbf{M}_{\theta\theta} \boldsymbol{\theta}_f^{n+1} \quad \mathbf{0} \right\}^T.
\end{aligned} \tag{4.12}$$

4.5 Discretization error and uncertainty estimation

4.5.1 Numerical errors

The main sources of numerical errors in the scientific computing simulations are round-off errors, statistical sampling errors, iterative and discretization errors (Roache, 2009, Oberkampf and Roy, 2010). Round-off errors arise due to the finite precision of the computers and they can be minimised by increasing the number of significant digits used to represent a number in the computations. Most of the physical systems are described by mathematical equations which may contain several non-linearities and iterative error is a consequence of solving the underlying non-linear equations. Statistical sampling errors exists in the solutions which are stochastic in nature. In that case, a number of realizations are needed to determine the mean quantities of interest. In the present work, statistical sampling errors are not considered. Discretization error is defined as difference between exact solution to the discrete equations and exact solution to the mathematical model (Oberkampf and Roy, 2010). In the numerical results presented in previous chapters, it is noticed that the round-off and iterative errors become dominant only around the range $10^{-9} - 10^{-10}$. Hence, in the present example, the contribution of round-off and iterative errors to numerical error is assumed to be negligible compared to discretization error.

4.5.2 Discretization error

Let ϕ be any scalar quantity of interest, for instance, temperature at a point, average Nusselt number, *etc.* If ϕ_n and ϕ_a are the numerical and analytical values of ϕ , respectively, the discretization error, ε , can be expressed in asymptotic regime as,

$$\varepsilon = |\phi_n - \phi_a| \approx c_h h^q + c_t \Delta t^\mu, \quad (4.13)$$

where, h is the element size of the domain, q is the order of convergence in space of the quantity of interest, μ is the order of convergence of time integration scheme, c_h and c_t are error constants. In equation (4.13), the terms $c_h h^q$, $c_t \Delta t^\mu$ quantifies the discretization errors in space and time, respectively. This relation can be used in *a priori* error estimation to establish the formal rates of convergence in space and time, when the analytical solution is known, as discussed in the previous chapters. The constants c_h and c_t depends on derivatives of the solution and their bounds can be proven only in the presence of smooth solutions. In the convergence results presented in the previous chapters, the considered analytical solutions are smooth and hence, rates of convergence could be established. However, in the present problem, the analytical solution is not known *a priori*. Therefore, to estimate analytical quantity of interest, *a posteriori* error estimation methods are employed. There had been plenty of study in *a posteriori* error estimation for finite element methods, see Babuška and Rheinboldt (1978), Whiteman (1994), Estep *et al.* (2000), Akin (2005).

One of the *a posteriori* error estimation methods is based on Richardson's extrapolation technique (Richardson, 1911). Originally, Richardson formulated a method which takes solutions from the two different nested solutions and compute a new solution that is one order more accurate than the original solution. Consider a second-order accurate scheme which has solutions with grid spacing h and $2h$. The discretization error equation can be expressed using Taylor's series as follows,

$$\begin{aligned} \phi_h &= \phi_a + a_1 h^2 + a_2 h^3 + \mathcal{O}(h^4), \\ \phi_{2h} &= \phi_a + a_1 (2h)^2 + a_2 (2h)^3 + \mathcal{O}(h^4), \end{aligned} \quad (4.14)$$

where ϕ_h, ϕ_{2h} are the discretized solutions for mesh h and $2h$, respectively and ϕ_a is the exact solution. Eliminating ϕ_a from equation (4.14) gives,

$$a_1 = \frac{\phi_{2h} - \phi_h}{3h^2} - \frac{7}{3}a_2h + \mathcal{O}(h^2). \quad (4.15)$$

From equation (4.15), substituting a_1 in (4.14) for the finer mesh, h , and solving for the analytical solution, ϕ_a gives,

$$\phi_a = \phi_h + \frac{\phi_h - \phi_{2h}}{3} + \frac{4}{3}a_2h^3 + \mathcal{O}(h^4). \quad (4.16)$$

The standard Richardson extrapolated solution is $\phi_h + \frac{\phi_h - \phi_{2h}}{3}$ and from equation (4.16), it is clear that the solution is third-order accurate, which is one order more than the original scheme. A more generalised extrapolated solution is $\phi_h + \frac{\phi_h - \phi_{rh}}{r_h^q - 1}$, where r_h is the refinement ratio and q is the order of convergence of the original scheme. This extrapolated solution can be used to estimate the discretization error for a given mesh as this solution has higher order of convergence than the original one. This method can be extended to space-time problems and it is discussed in Richards (1997). One of the important assumptions of this method is that the solutions that are considered to compute the extrapolated estimate should be in asymptotic range. Uniform meshes, nested mesh refinement and smooth solutions are amongst the other assumptions. In the case of space-time extrapolation, mesh sizes and time steps should be chosen in such a way that the errors from time and space should be of same order of magnitude. In the GFRP problem considered here, some of the mentioned assumptions cannot be verified easily. In addition, the geometry with the straight corners introduce singularities in the solution of heat flux. In the presence of singularities and discontinuities, the reliability and applicability of Richardson's extrapolated error estimated is limited as the order of convergence of the quantity of interest will lose the formal order of convergence. In general, this loss of convergence can be noticed clearly in the vicinity of the singularity. Moreover, in the present problem, non-monotonic convergence in space and time is observed for various quantities of interest.

To address the mentioned drawbacks, the method proposed in Eça and Hoekstra (2014) is used in the present study. Following their work, the discretization error of any quantity of interest, for either CG(Q_{k+1})-HDG(Q_k) or CG(Q_{k+1})-CG($Q_{k+1}Q_k$) coupled model using BDF μ time integration scheme, can be stated as follows,

$$\varepsilon_i(\phi) = \phi_i - \phi_a \approx c_{h,1}h_i + \dots + c_{h,k+2}h_i^{k+2} + c_{t,1}\Delta t_i + \dots + c_{t,\mu}\Delta t_i^\mu, \quad (4.17)$$

where $k+2$ and μ are the orders of convergence in space and time, respectively. The unknowns in equation (4.17) are computed using the non-linear least-squares fit of the solutions with various mesh sizes and time steps. It is generally recommended to use more grid points than unknowns in equation (4.17) to obtain more reliable data fits.

If n_d is the number of different solutions in space and time, the least-squares functional can be defined as follows,

$$Q_{LS}(\phi_a, c_{h,1}, \dots, c_{h,k+2}, c_{t,1}, \dots, c_{t,\mu}) = \sqrt{\sum_{i=1}^{n_d} w_i \left(\phi_i - (\phi_a + c_{h,1}h_i + \dots + c_{h,k+2}h_i^{k+2} + c_{t,1}\Delta t_i + \dots + c_{t,\mu}\Delta t_i^\mu) \right)^2}, \quad (4.18)$$

where w_i in the equation (4.18) is the weight of the observation and it can be defined as follows,

$$w_i = \frac{\frac{1}{h_i} + \frac{1}{\Delta t_i}}{\sum_{i=1}^{n_d} (\frac{1}{h_i} + \frac{1}{\Delta t_i})}. \quad (4.19)$$

In equation (4.19), w_i depends on the grid spacing, h_i , and time step, Δt_i . Observations with smaller time step and smaller grid spacing will have bigger weight than the rest of the observations. This weighted approach is advantageous in the case of non-monotonic convergence behaviour as it enables to compute the discretization error by giving more weight to finer grid solutions than coarser ones. The definition of weights (4.19) guarantees the sum of weights to be unity.

Finally, standard deviation of the data points from the regression analysis can be defined as,

$$\hat{\sigma}_\varepsilon = \sqrt{\frac{\sum_{i=1}^{n_d} n_d w_i \left(\phi_i - (\phi_a + c_{h,1} h_i + \dots + c_{h,k+2} h_i^{k+2} + c_{t,1} \Delta t_i + \dots + c_{t,\mu} \Delta t_i^\mu) \right)^2}{n_d - (k + 2 + \mu + 1)}}. \quad (4.20)$$

However, in the present work the least-squares functional without weights is also considered. In this case, $w_i = 1$ in equation (4.18) and $n_d w_i = 1$ in equation (4.20). For each quantity of interest, least-squares fits with and without weights are computed and the model that produces smaller standard deviation is chosen to compute the uncertainty.

4.5.3 Uncertainty estimation

The computed discretization error and standard deviation can be used to compute the numerical uncertainty of the computations. If ϕ_a is the exact solution and U_ϕ is the uncertainty, then the confidence interval of an observation ϕ_i is,

$$\phi_i - U_\phi \leq \phi_a \leq \phi_i + U_\phi. \quad (4.21)$$

The uncertainty is estimated based on the quality of the data fit. To measure the quality, a data range parameter is defined as follows,

$$\Delta_\phi = \frac{\max(\phi_i) - \min(\phi_i)}{n_d - 1}. \quad (4.22)$$

The quality of the fit is judged using the standard deviation, $\hat{\sigma}_\varepsilon$, and data range parameter, Δ_ϕ . If $\hat{\sigma}_\varepsilon \leq \Delta_\phi$, the error estimate is assumed to be good, while $\hat{\sigma}_\varepsilon > \Delta_\phi$ results in a bad error estimate. Based on this, the following expressions are used to compute the uncertainty interval:

$$U_\phi(\phi_i) = \begin{cases} 1.25 \varepsilon_i(\phi) + \hat{\sigma}_\varepsilon + |r|_i & \text{if } \hat{\sigma}_\varepsilon \leq \Delta_\phi, \\ 3 \frac{\hat{\sigma}_\phi}{\Delta_\phi} (\varepsilon_i(\phi) + \hat{\sigma}_\varepsilon + |r|_i) & \text{if } \hat{\sigma}_\varepsilon > \Delta_\phi, \end{cases} \quad (4.23)$$

where the values 1.25 and 3 are the safety factors chosen based on Grid Convergence Index (GCI) (Roache, 2009). In equation (4.23), $\varepsilon_i(\phi)$ is discretization error and $|r|_i$ is the absolute value of the residual of the fit at the solution, i . As it is clear from equation (4.23), the uncertainty accounts not only discretization error, but also the error of least-square fit of the solution and the standard deviation, which are the consequence of scatter in the data. Using the equation (4.23), the uncertainty of any solution can be computed. However, in this work, only the uncertainty corresponding to the solution of finest grid space and smallest time step is considered and the same is presented in the results.

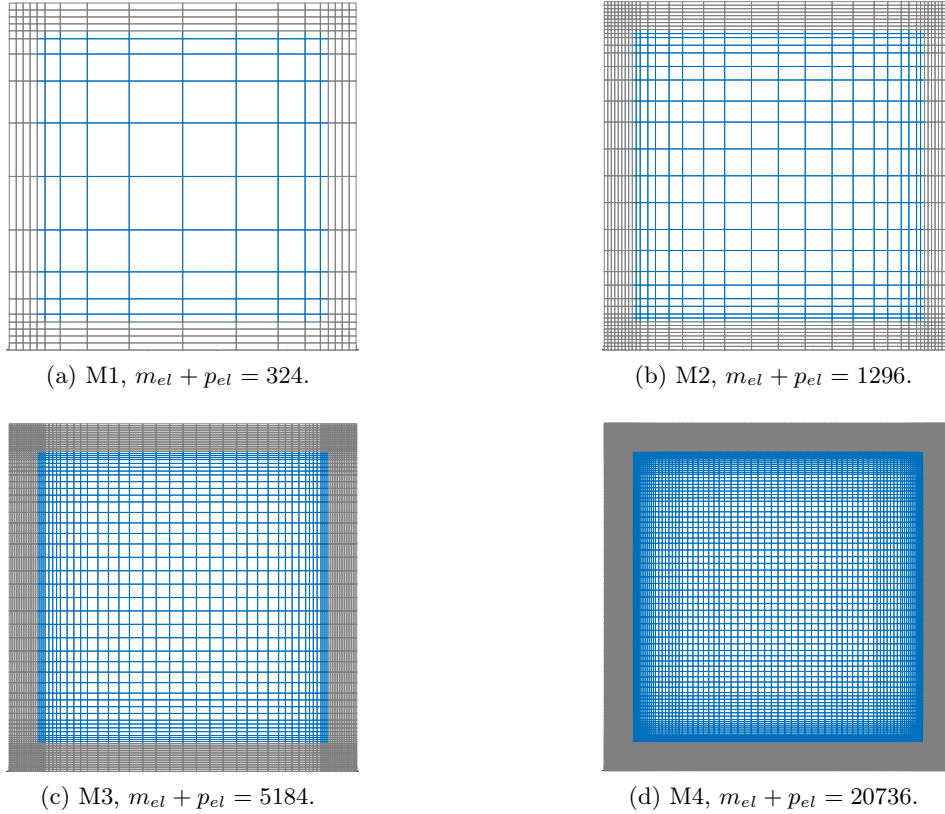


Figure 4.5: Nested structured meshes used in the analysis of the GFRP problem.

4.6 Numerical results

4.6.1 Problem data

As stated earlier, a GFRP cross section with straight and curved corners in the cavity is considered in the present work. The cross section with the straight corners is discretized using structured meshes, while curved corners geometry needs unstructured meshes. Figures 4.5 and 4.6 show different structured and unstructured meshes along with their designation, respectively. Both structured (M1 to M4) and unstructured (UM1 and UM2) meshes are nested. The largest element size of meshes M1 to M4 are $0.0155/\{2^0, 2^1, 2^2, 2^3\}$, respectively. The time steps used in the analysis are $0.0625/\{2^0, 2^1, 2^2, 2^3\}$ and the problem is analysed for a total time of $T = 400$ sec.

As evident from the figs. 4.5 and 4.6, quadrilateral elements and triangular elements are used for structured and unstructured meshes, respectively. In all the numerical results corresponding to CG-HDG coupled model, the stabilization constants, τ_u and τ_θ are taken as 1. At time $t = 0$, the temperature of GFRP section and the air in the cavity is assumed to be at reference temperature, $\theta_0 = 24^\circ\text{C}$, and fluid starts at rest, *i.e.*, $\mathbf{u}^0 = \mathbf{0}$.

4.6.2 Development of flow

In this section, the development of the flow pattern and corresponding temperature distribution are discussed. Numerical results of the coupled CG(Q_2)-HDG(Q_1) model are presented

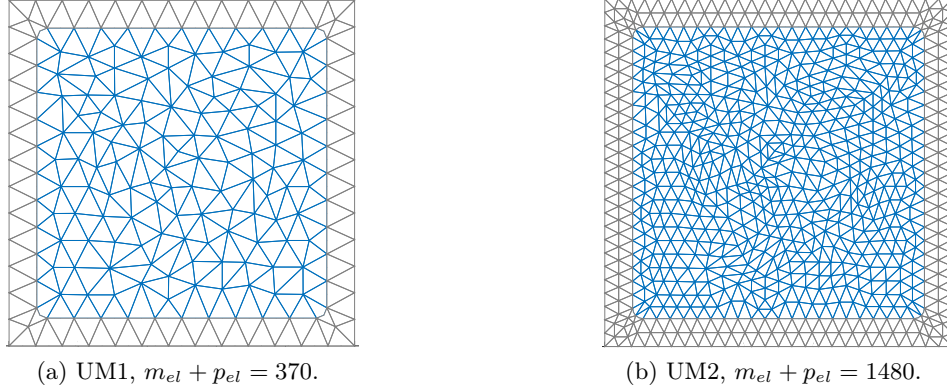


Figure 4.6: Nested unstructured meshes used in the analysis of the GFRP problem.

using mesh M4 and a time step of $0.0625/2^3$ sec. The isolines of temperature and velocity are shown for 4 different time instances in fig. 4.7. At $t = 50$ sec, it can be noticed that the dominant mechanism of heat transfer is conduction in the GFRP section. This is evident from the velocity isolines in fig. 4.7b, where the velocity is relatively insignificant. It can also be observed the initiation of formation of convective cells in the GFRP cavity. At $t = 150$ sec, the formation of a thermal plume is noticed clearly from fig. 4.7c. A symmetric structure with two counter rotating vortices is developed in the cavity of the GFRP. Due to the large difference between the conductivities of GFRP and air, a thermal boundary layer is created near the lateral walls of the cavity. As the temperature is further increased, the convection velocity also increases in the cavity as shown in fig. 4.7f. The flow is still symmetric at this time with convection becoming more significant with the increased buoyancy force. At around $t = 240$ sec, the symmetric structure in velocity and temperature are lost due to instabilities in the vortex structures. From this time instance, the center of the convective vortices move back and forth vertically with a type of oscillatory behaviour. Figure 4.7g shows the loss of symmetry in velocity at $t = 385$ sec. It can also be observed that the vortex centers are lower compared to ones in fig. 4.7f. This type of oscillatory behaviour can be illustrated by considering the temperature at the center $(0.05, 0.05)$ of the cavity and average Nusselt number along the lower edge of the cavity, which is represented in red in fig. 4.9. The average Nusselt number is computed as follows,

$$\overline{Nu} = -\frac{1}{\theta_a - \theta_0} \int_{0.008}^{0.092} \frac{\partial \theta_s}{\partial x_2} ds, \quad (4.24)$$

where θ_a can be computed from ISO profile in equation (4.3) at 400 sec and θ_0 is the initial reference temperature of 24°C . Figure 4.8 shows the variation of temperature at $(0.05, 0.05)$ and average Nusselt number, \overline{Nu} , along the time. It can be observed that the flow is in a type of oscillatory state with the average values of the quantities increasing with time. This type of oscillatory behaviour is reported in the work Stella and Buchignani (1999), in the context of Rayleigh–Bénard convection flow. The oscillatory behaviour of the solution makes it very difficult to access the numerical accuracy of the solution.

From the engineering point-of-view, two main quantities of interest are considered in this problem namely, temperature in GFRP section and average Nusselt number along the bottom edge of the cavity. The distribution of thermocouples to monitor the temperature in the GFRP cross-section in the experimental set up is shown in fig. 4.9 and the corresponding co-ordinate

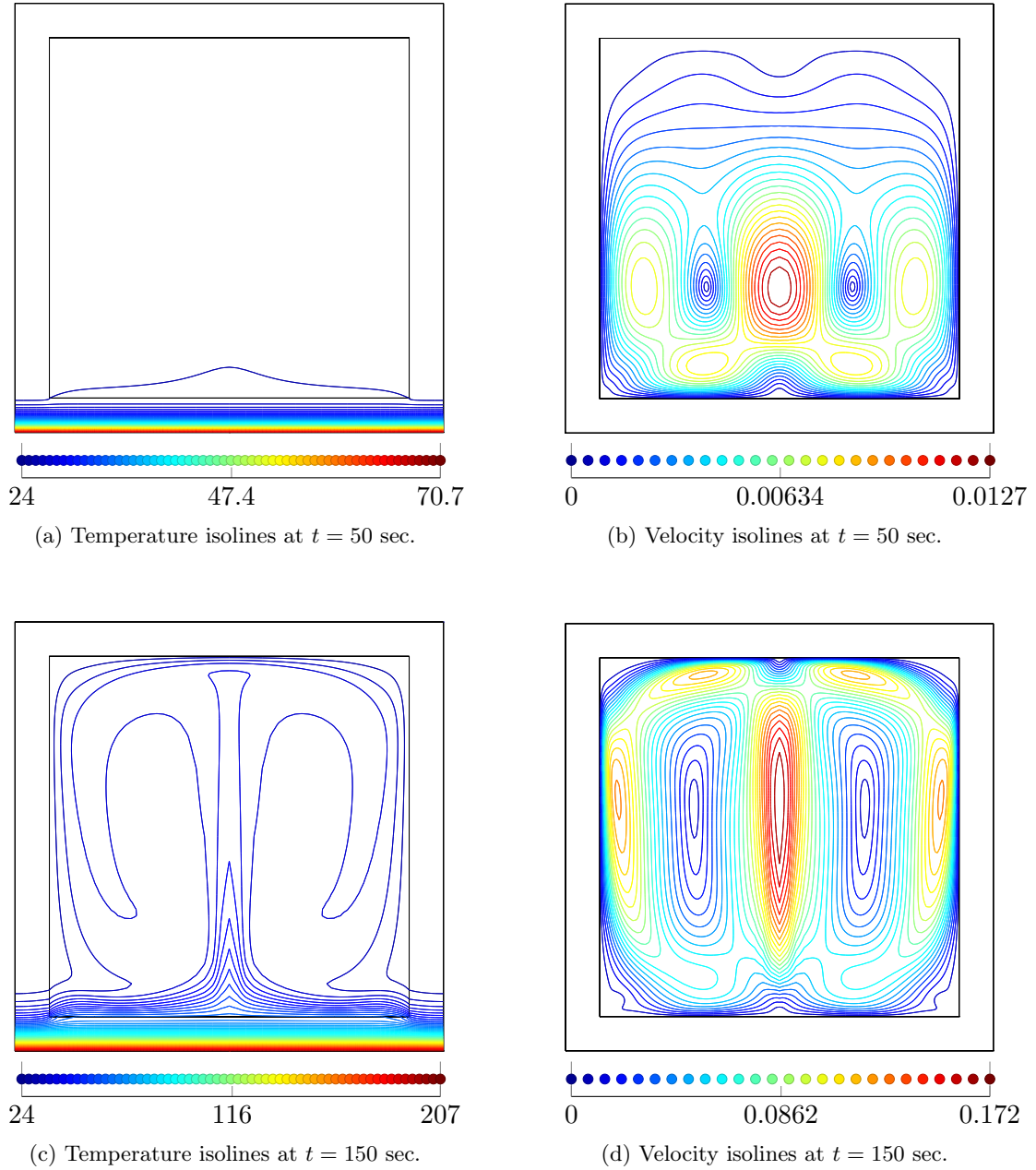


Figure 4.7: GFRP problem: Evolution of temperature and velocity in the cavity at $t = 50, 150, 230$ and 385 sec using coupled $\text{CG}(Q_2)\text{-HDG}(Q_1)$ model. Post-processed solutions are shown in Ω_f .

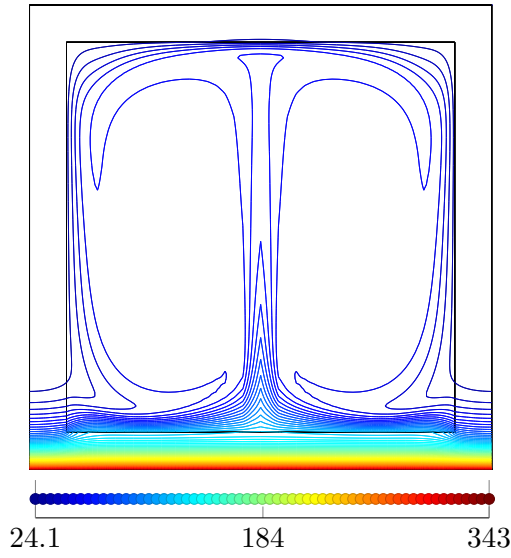
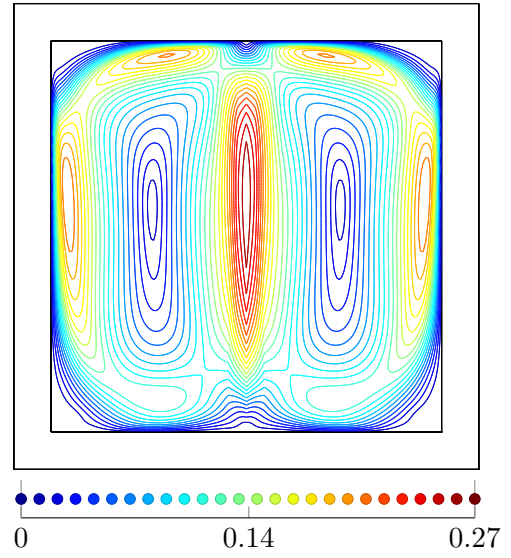
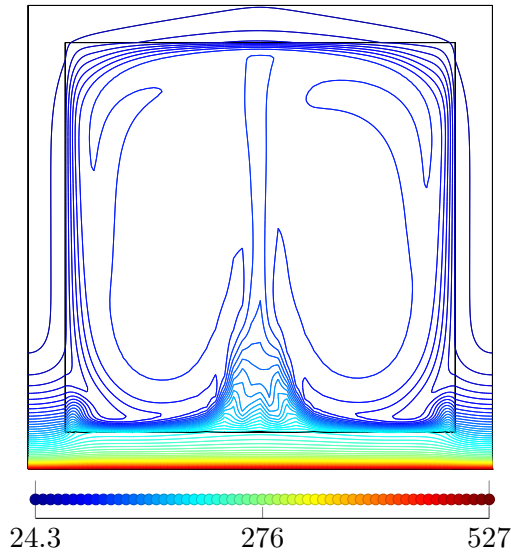
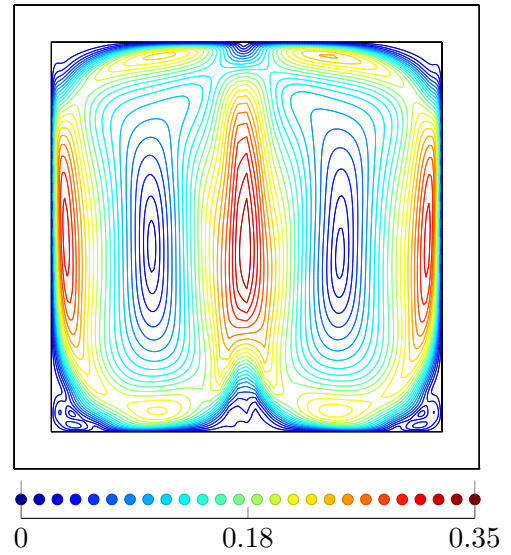
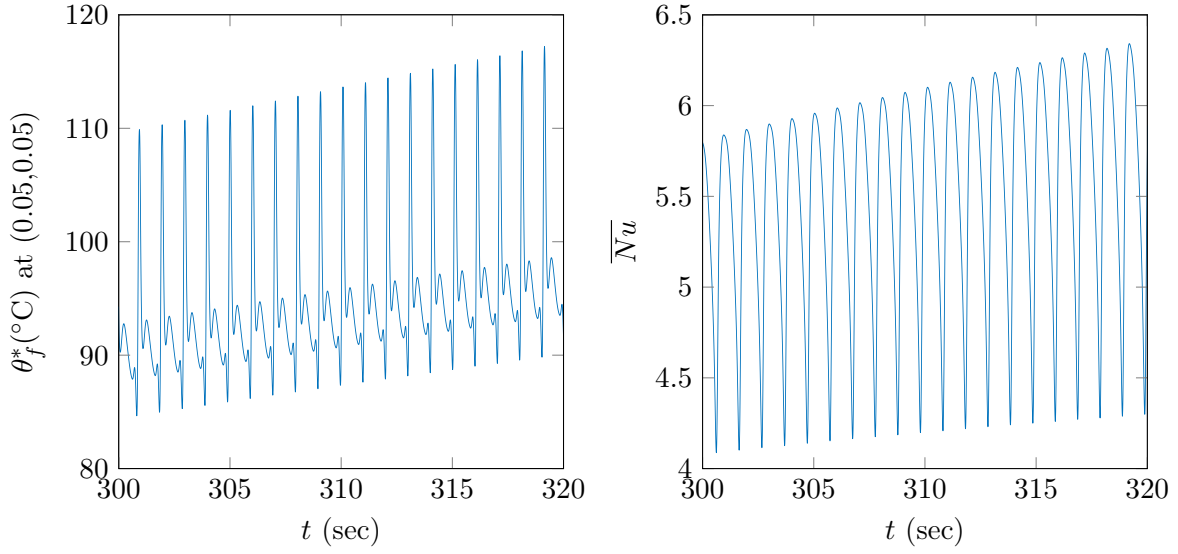
(e) Temperature isolines at $t = 230$ sec.(f) Velocity isolines at $t = 230$ sec.(g) Temperature isolines at $t = 385$ sec.(h) Velocity isolines at $t = 385$ sec.

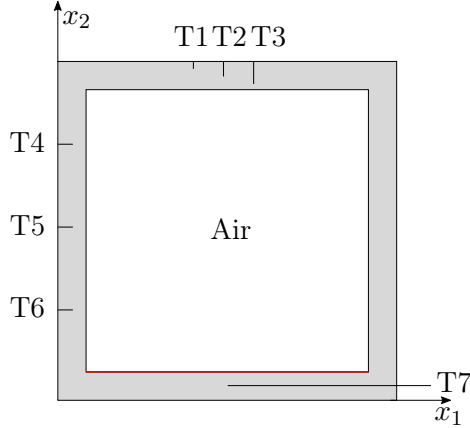
Figure 4.7: GFRP problem: Evolution of temperature and velocity in the cavity at $t = 50, 150, 230$ and 385 sec using coupled $\text{CG}(Q_2)$ -HDG(Q_1) model. Post-processed solutions are shown in Ω_f . Continuation.



(a) Post-processed temperature in the cavity at (0.05, 0.05). (b) Average Nusselt number along $x_2 = 0.008$.

Figure 4.8: GFRP problem: variation of temperature and Nusselt numbers with time.

positions are presented in table 4.1. Thermocouples T1, T2 and T3 are present in the top flange, T4, T5 and T6 in the web and T7 in the bottom flange (Morgado *et al.*, 2013b).



Thermocouple	$x_1(\text{m})$	$x_2(\text{m})$
T1	0.040	0.098
T2	0.050	0.096
T3	0.060	0.094
T4	0.004	0.075
T5	0.004	0.050
T6	0.004	0.025
T7	0.050	0.004

Figure 4.9: Locations of thermocouples in GFRP cross-section. Table 4.1: Positions of different thermocouples.

The average Nusselt number quantifies the amount of energy being added to the system with time. Since, it shows an oscillatory behaviour as shown in fig. 4.8b, the average Nusselt number is averaged in time in order to be able to compare the numerical results between different coupled models. This averaging in time is done by using method of moving average, also known as rolling mean. A simple moving average method is used in the present context and it is applied to average Nusselt number which can be expressed as follows,

$$\overline{Nu}_{ma}(t_j) = \frac{1}{2w+1} \sum_{k=j-w}^{j+w} \overline{Nu}(t_k), \quad (4.25)$$

where $\overline{Nu}_{ma}(t_j)$ is the moving average value of Nusselt number at time step j and w is the length of the window on either side of j^{th} time step. Hence, a central mean is obtained by taking the average of w time steps in forward and backward directions of time step j . In the present work, the parameter w is chosen to cover 1 second of time and hence, for different time steps w changes. This method is analogous to using a low-pass filter to remove the higher frequencies. This procedure is done successively until a smooth averaged variation in time obtained. It is noticed that in the present case, applying this moving average method for 5 times removes all the high frequency data and a smooth trend in time is obtained.

4.6.3 Estimation of uncertainty

In this section, uncertainties in temperature of thermocouples in GFRP cross-section and average Nusselt number are computed. This study is performed for CG(Q_2)-CG(Q_2Q_1) and CG(Q_2)-HDG(Q_1) coupled models which have optimal order of convergence of 3 for velocity and temperature and 2 for pressure and heat flux. For each mesh M1, M2, M3 and M4, computations with four different time steps of $0.0625/\{2^0, 2^1, 2^2, 2^3\}$ are considered. This gives 16 systematically refined solutions in space and time. However, it is noticed from the results that the meshes M1 and M2 are clearly not in asymptotic limit for the considered degree of approximation. Hence, results corresponding to M1 and M2 are discarded from the analysis, which leaves with 8 different solutions. If θ_i^c is the temperature at any thermocouple c , $\overline{Nu}_{ma,i}(t_j)$ is the average Nusselt number at any time step j , then discretization errors in the corresponding quantities for any solution i can be written as,

$$\begin{aligned} \varepsilon_i(\theta^c) &= \theta_i^c - \theta^c && \approx c_{h,1} h_i + c_{h,2} h_i^2 + c_{h,3} h_i^3 + c_{t,1} \Delta t_i + c_{t,2} \Delta t_i^2, \\ \varepsilon_i(\overline{Nu}_{ma}(t_j)) &= \overline{Nu}_{ma,i}(t_j) - \overline{Nu}_{ma}(t_j) && \approx c_{h,1} h_i + c_{h,2} h_i^2 + c_{t,1} \Delta t_i + c_{t,2} \Delta t_i^2, \end{aligned} \quad (4.26)$$

for $i = 1, \dots, 8$. Since, flux is only second-order accurate, the term $c_{h,3} h_i^3$ is eliminated in the expression for $\varepsilon_i(\overline{Nu}_{ma})$. Therefore, there are 6 unknowns in error equation corresponding to temperature, while only 5 in the case of Nusselt number. In any case, there are 8 different solutions, which are more than enough to compute the unknown variables from both quantities. In all the results, it is noticed that the terms corresponding to first-order in space and second-order in time, *i.e.*, $c_{h,1} h_i$ and $c_{t,2} \Delta t_i^2$, are significant compared to other terms. This suggests that the convergence in space is restricted to order 1 due to the presence of singularities, whereas optimal convergence order of 2 is recovered in time.

Firstly, the results of evolution of temperature in different thermocouples and moving average of space averaged Nusselt number, \overline{Nu}_{ma} , are presented for CG(Q_2)-CG(Q_2Q_1) and CG(Q_2)-HDG(Q_1) coupled models for the mesh M4 and time step of $0.0625/2^3$ sec in fig. 4.10. These solutions correspond to the finest grid size and smallest time step and hence, are the most accurate solutions available for both coupled models. From the fig. 4.10, it is evident that both coupled models produced very similar results. In fig. 4.10a, a slight difference between temperatures in T1, T2 and T3 can be noticed, while the results are very similar in figs. 4.10b and 4.10c, *i.e.*, temperatures in web and bottom flange. The temperatures in top flange *i.e.*, T1, T2 and T3 are observed to be higher than in the web. The temperature isolines in fig. 4.7 suggest that the heat flux entering the top flange is higher compared to the web and hence, higher temperatures. It is noticed that the temperature evolution of T7 is the same for both coupled models, as presented in fig. 4.10c. This is to be expected, as both coupled models use CG to discretize GFRP part and the effect of fluid convection on this thermocouple is very small. Finally, the moving average of the average Nusselt number of

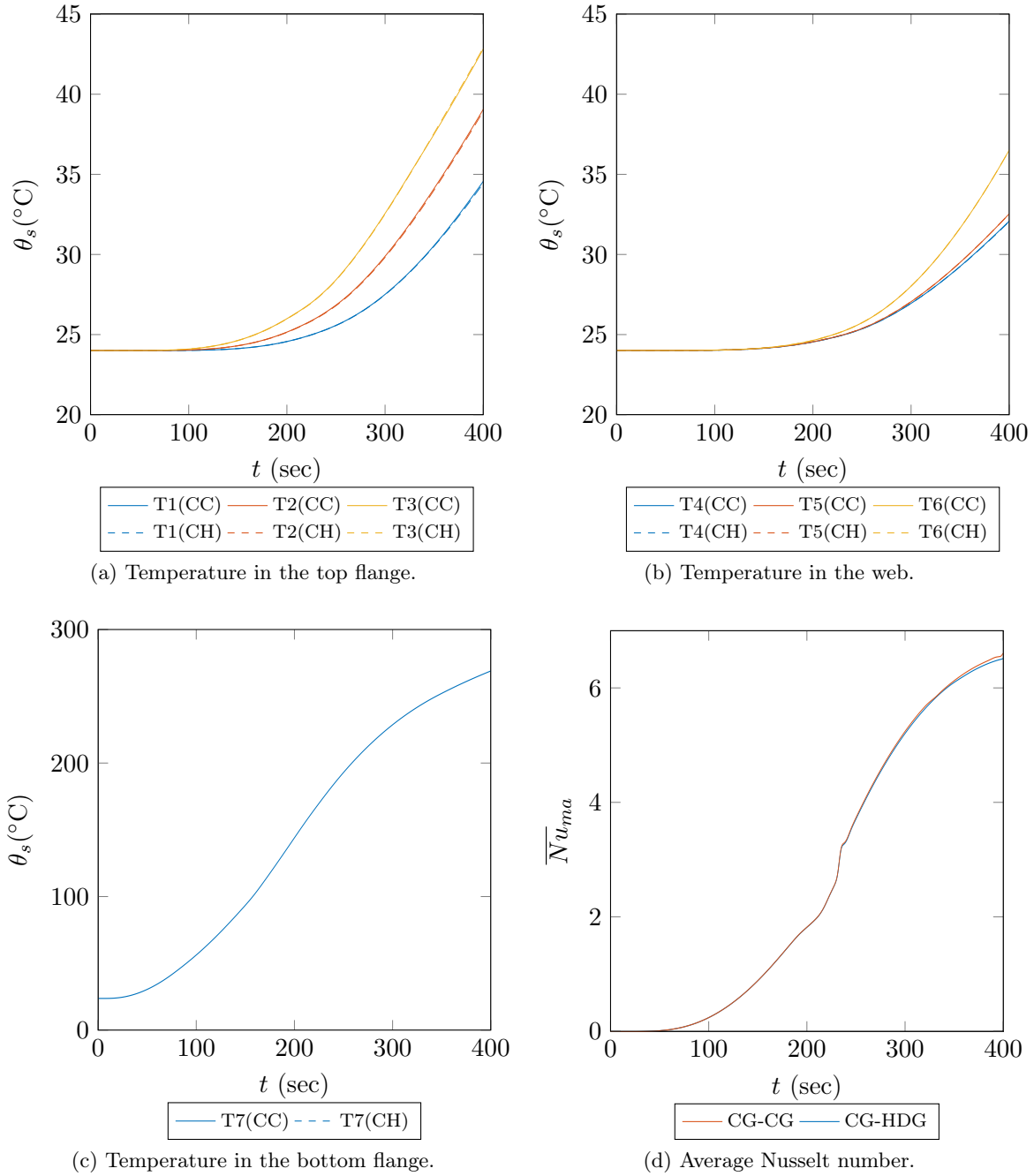


Figure 4.10: GFRP problem: evolution of temperature in thermocouples and average Nusselt number with time using mesh M4 and time step used is $0.0625/2^3$. CC and CH denotes CG-HDG and CG-CG models, respectively.

Table 4.2: RMS values of errors in temperature for different thermocouples and average Nusselt number for the considered coupled models.

Model	Thermocouples (in °C)							Nusselt Number
	T1	T2	T3	T4	T5	T6	T7	
CG-HDG	0.3011	0.0519	0.1542	0.0171	0.0204	0.0205	0.0510	0.0715
CG-CG	0.0645	0.6023	3.4795	0.0573	0.0124	0.0098	0.0261	0.1221

both CG-HDG and CG-CG coupled models are in very good agreement. The kink observed in the average Nusselt number plot in fig. 4.10d around $t = 225$ sec coincides with the time where the instabilities appear in the convective cells and the flow becomes oscillatory.

Figure 4.11 shows the estimated uncertainty in the temperature of the thermocouples and the average Nusselt number for both coupled models. The time period of $t = 200 - 400$ sec is considered to present the uncertainty interval as solutions tend to change more with refinement after $t = 200$ sec. One common trend observed in most of the data points is that uncertainty is very small and most of the times, intervals from coupled CG-HDG and CG-CG models overlap. This concludes that the considered coupled models give very similar results in the present context. In fig. 4.11a, it can be noticed that the error in thermocouple T3 of CG-CG coupled model produces a larger uncertainty compared to all other quantities. This might be due to the poor non-linear least-squares fit and this error interval can be deemed to be unreliable.

Figure 4.11d shows the moving average of the average Nusselt number for both coupled models and they are in good agreement. However, the largest uncertainty interval obtained in the case of average Nusselt number for both coupled models is around $t = 240$ sec. As stated earlier, this time instance is in transition phase, where an oscillatory behaviour starts to develop in the cavity. Once, the flow develops into oscillatory behaviour, the uncertainty interval is very small for both coupled models. The time instance at which the oscillations start depends on the numerical perturbations which depends on mesh, time step, *etc.* Hence, the uncertainty is larger in this phase and once the flow settles into oscillatory behaviour, both models converge to the same solution.

It is very difficult to assess the quality of the solutions based on the uncertainties presented in fig. 4.11. Hence, a Root Mean Square (RMS) value of error is computed for the thermocouple temperatures and average Nusselt number to compare the average errors of both coupled models. Table 4.2 shows the RMS values of errors for different thermocouples and average Nusselt number for coupled CG-HDG and CG-CG models. It can be observed from the error values of thermocouple temperatures that CG-HDG is better in some cases, while CG-CG produces smaller errors in other cases. Nevertheless, the errors are very low, *i.e.*, within 1% for most of the results for both coupled models. Thermocouple T3 of CG-CG coupled model has the highest RMS value of error. As stated earlier, this might be due to unreliable data fitting and hence, should be of least importance. Even in the case of average Nusselt number, the error is within 1% of maximum Nusselt number in time for CG-HDG model, while it is within 2% for coupled CG-CG model. Hence, it can be concluded that both models give the results with very close accuracy.

Now, it is interesting to compare the CPU times taken by the linear solvers for both coupled models. Table 4.3 shows the total time taken by the linear solver and the number of solves that both coupled models needed for the same mesh and time step. Even though,

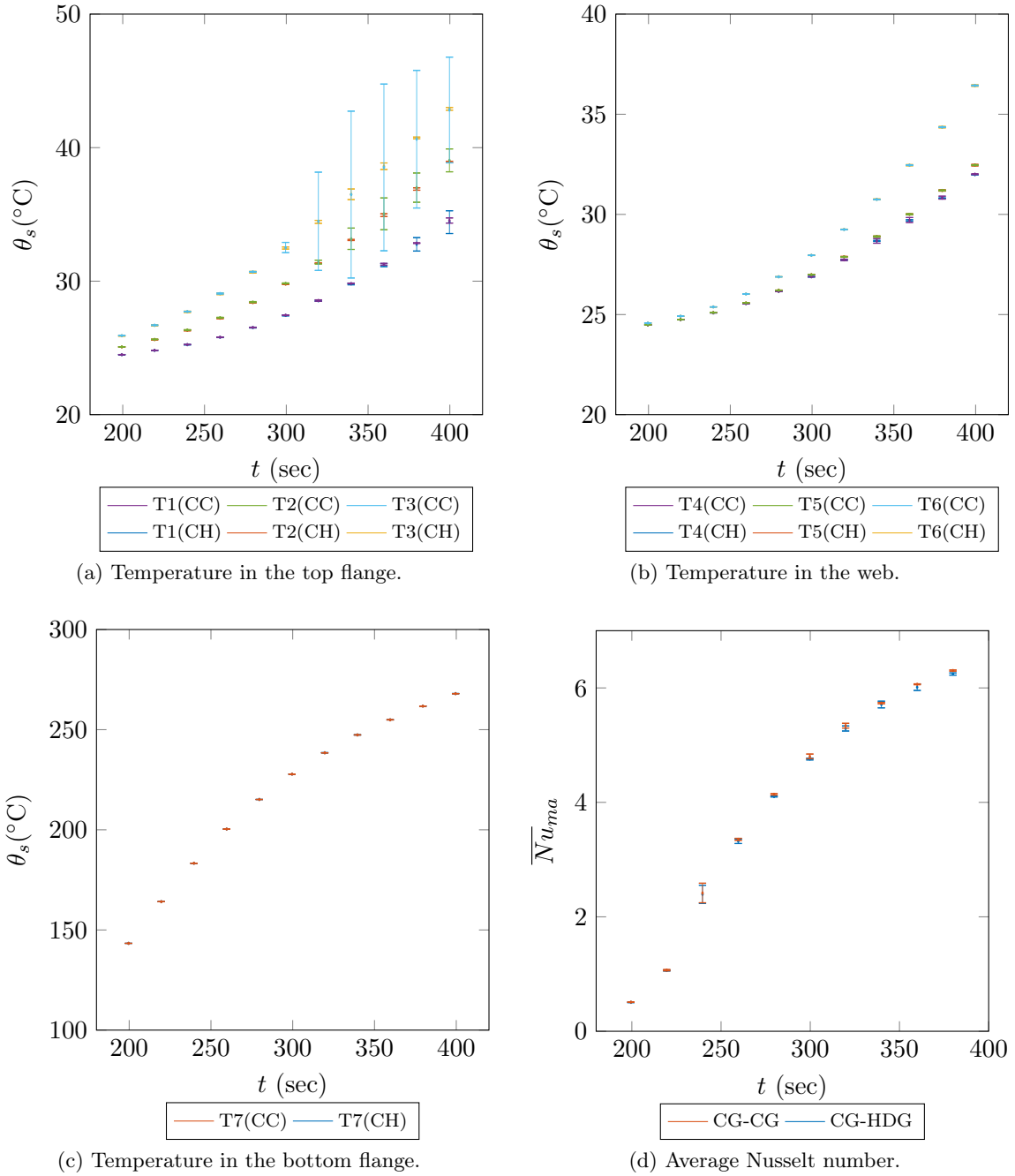


Figure 4.11: GFRP problem: uncertainty in temperature in thermocouples and average Nusselt number with time using mesh M4 and time step used is $0.0625/2^3$. CC and CH denotes CG-HDG and CG-CG models, respectively.

Table 4.3: Computational details for coupled CG(Q_2)-CG(Q_2Q_1) and CG(Q_2)-HDG(Q_1) models for GFRP problem with mesh M4 and time step of $0.0625/2^3$.

Model	ndof	nnz	Total CPU time for linear solver (in hours)	Number of solves	CPU Time/solve (in seconds)
CG-HDG	1 28 064	45 93 485	567 hr	1 74 687	11.69 sec
CG-CG	1 09 028	46 03 053	854 hr	1 74 716	17.61 sec

coupled CG-HDG model has more DOFs compared to coupled CG-CG model, the number of non-zero entries in the global matrix is marginally lower for CG-HDG model. Coupled CG-HDG is clearly favourable when number of solves and total CPU time for linear solver is considered. CG-HDG model is at least 1.5 times faster than CG-CG model when CPU time for solving one system is considered. In the examples presented in the previous chapters, CG-CG model is very competitive at low-degree approximations when the same mesh is considered. However, it can be concluded that CG-HDG coupled model can be competitive, even at the low-degree approximations, when highly refined meshes are used.

4.6.4 High-order results

In this section, results obtained using high-order elements using coupled CG(Q_{k+1})-HDG(Q_k) and CG(Q_{k+1})-CG($Q_{k+1}Q_k$) are presented. The mesh size and degree of approximation are chosen in such a way that the approximate total number of nodes are constant. Following this idea, mesh M2 with elements of degree $k = 4$ has approximately the same number of nodes of mesh M1 with degree $k = 8$. Hence, mesh M1 is considered with coupled CG(Q_8)-HDG(Q_7) and CG(Q_8)-CG(Q_8Q_7) models. In the same way, mesh M2 is chosen with coupled CG(Q_4)-HDG(Q_3) and CG(Q_4)-CG(Q_4Q_3) models. All the results presented use a constant time step of $0.0625/2^3$, which is the smallest time step in this analysis.

Figure 4.12 presents the results of temperatures in the thermocouples and the average Nusselt number for coupled CG(Q_{k+1})-HDG(Q_k) model for different degrees of approximations. The results from both computations are very close to each other in all the quantities. It can be observed from plots in fig. 4.12 that the computations with CG(Q_4)-HDG(Q_3) model slightly under predicts compared to numerical results of CG(Q_8)-HDG(Q_7) model. Moreover, the uncertainty estimates computed from the earlier study are used here to verify the results and it is noticed that the results of the two higher order approximations, $k = 3$ and $k = 7$, are in good agreement with the established confidence intervals. It is worth noting that these higher order results have a numerical uncertainty of their own. From the plots, it can be inferred that these uncertainties either match the computed estimates or overlap the part of the interval. Figure 4.13 presents the results of the coupled CG(Q_{k+1})-CG($Q_{k+1}Q_k$) model with high-degree approximations. In this case, the results from both degrees of approximation practically coincide. As shown in fig. 4.13, it is very difficult to distinguish the results from the plots. The uncertainty estimates of CG-CG model from earlier study are shown in the plots to verify them. It is noticed that all the obtained results are well within the uncertainty intervals for both set of results.

The notable difference in the results between two degrees of approximations in the coupled CG-HDG models can be observed in thermocouple T3 and the average Nusselt number as shown in figs. 4.12a and 4.12d. These two quantities have a strong dependence on the

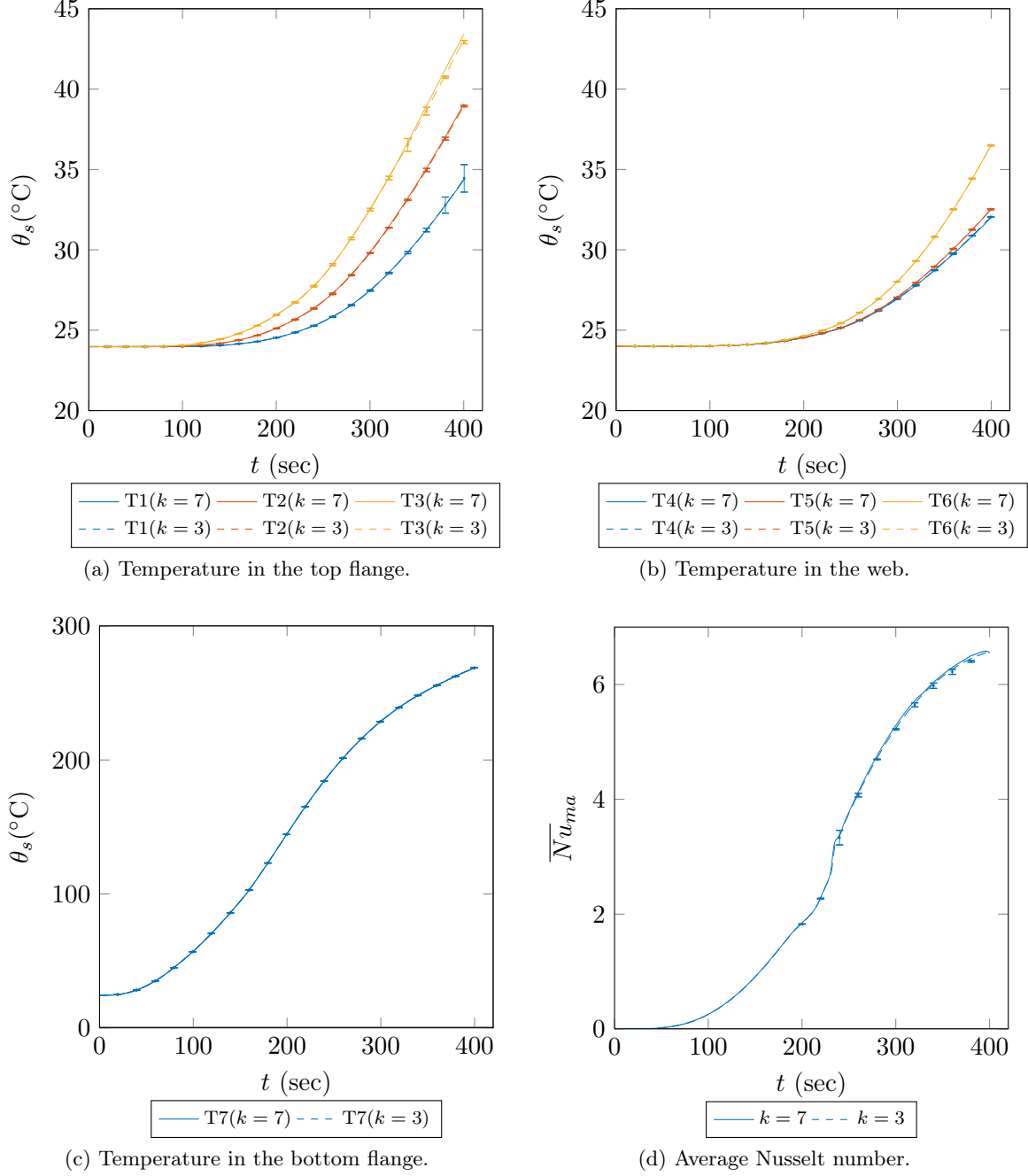


Figure 4.12: GFRP problem: evolution temperature in thermocouples and average Nusselt number with time with high-order elements using coupled CG(Q_{k+1})-HDG(Q_k) model. Solid lines and dashed lines correspond to M1 and M2 meshes, respectively and time step used is $0.0625/2^3$.

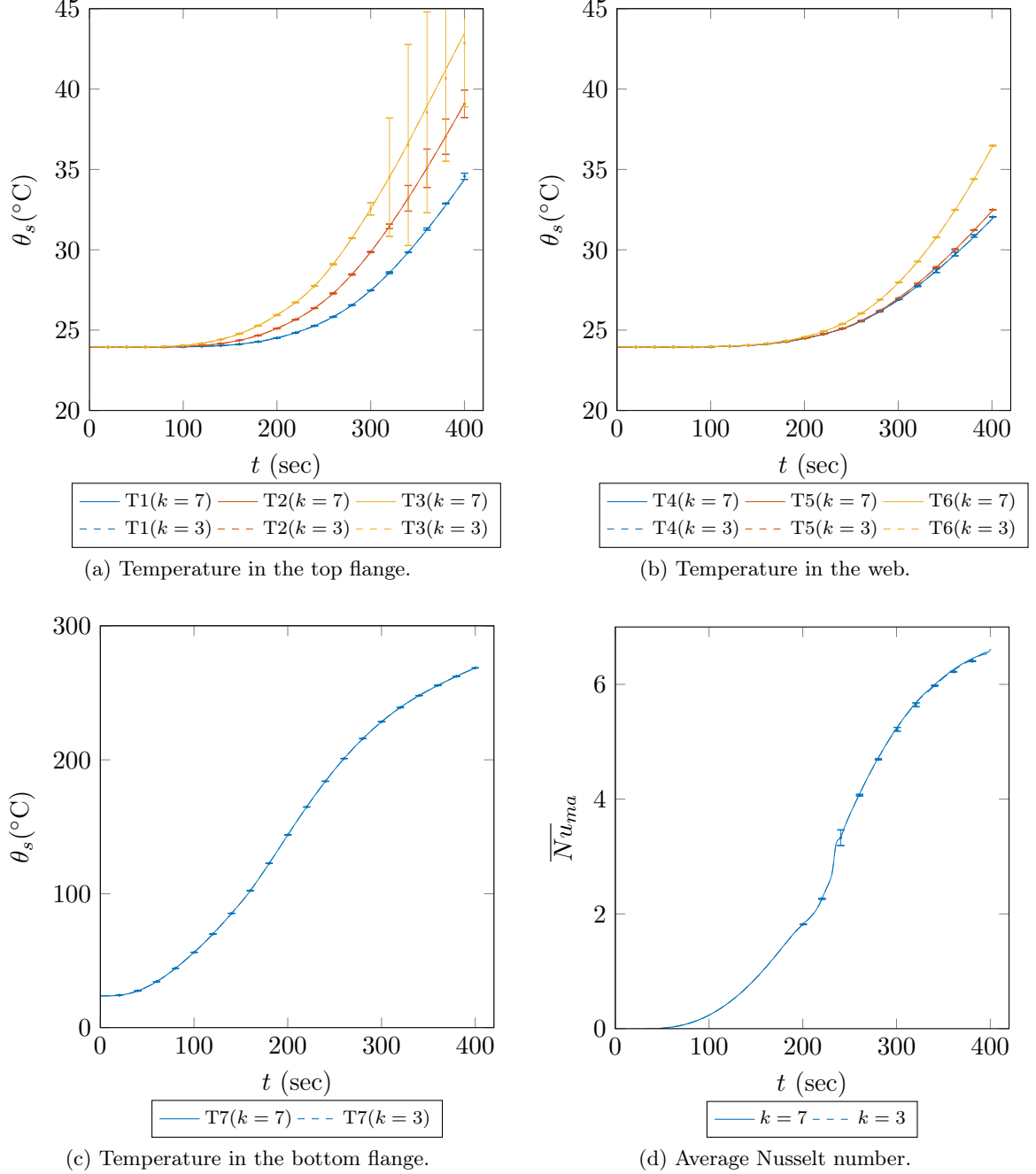


Figure 4.13: GFRP problem: evolution temperature in thermocouples and average Nusselt number with time with high-order elements using coupled $\text{CG}(Q_{k+1})\text{-CG}(Q_{k+1}Q_k)$ model. Solid lines and dashed lines correspond to M1 and M2 meshes, respectively and time step used is $0.0625/2^3$.

Table 4.4: Computational details for coupled CG(Q_{k+1})-HDG(Q_k) and CG(Q_{k+1})-CG($Q_{k+1}Q_k$) models for GFRP problem with mesh M1 ($k = 7$) and mesh M2 ($k = 3$) with a time step of $0.0625/2^3$.

Model	ndof	nnz	Total CPU time for linear solver (in hours)	Number of solves	CPU Time/solve (in seconds)
CG(Q_8)-HDG(Q_7)	9 188	10 48 569	62.0 hr	1 74 696	1.277 sec
CG(Q_4)-HDG(Q_3)	17 200	10 98 805	91.4 hr	1 74 538	1.885 sec
CG(Q_8)-CG(Q_8Q_7)	10 232	14 80 905	84.3 hr	1 74 683	1.731 sec
CG(Q_4)-CG(Q_4Q_3)	17 724	13 90 933	123 hr	1 74 654	2.539 sec

convection of the air inside the cavity and hence, the difference in results can be attributed to differences in the fluid solutions between both results. This could be due to the existence of singularities introduced by the geometry at the four corners of the cavity, as stated earlier in the chapter. The consequences of the singularities on the proposed CG-HDG model is not clear, as all the examples considered until this point in the work have smooth solutions. This coupled formulation needs a thorough mathematical analysis to understand the effect of the singularities in the solution. As this is out of scope of the present thesis, it can be referred as one of the future developments of the present work.

Finally, CPU times for linear solvers are compared along with other important computational quantities. Table 4.4 summarises the comparison between the two sets of the results with coupled CG-HDG and CG-CG models. First, comparing within CG-HDG and CG-CG models infers that high-order computations take lower CPU time for linear solver than low-order ones. Even though, there is significant difference in **ndof** and the number of non-zero entries, **nnz**, are approximately the same for each degree of approximation of the corresponding coupled models. However, the comparison of coupled CG-HDG model with coupled CG-CG model for corresponding degrees of approximations suggest that CG-HDG is more efficient for both computations for the meshes M1 ($k = 7$) and M2 ($k = 3$) with the ratio of CPU times around 0.7 favouring CG-HDG model. Hence, the conclusion inferred from study in Chapter 2 with benchmark example holds in this present practical application problem.

4.6.5 Results with rounded corners in cavity

This section presents the results performed using rounded corners in the cavity of the GFRP profile cross-section. Meshes UM1 and UM2 are used in the analysis along with the time step of $0.0625/2^3$. This type of geometry ensures the absence of singularities and it is more realistic geometric configuration. However, unstructured meshes are considered. Since the corners are rounded, the average Nusselt number is redefined as follows,

$$\overline{Nu} = -\frac{1}{\theta_a - \theta_0} \int_{0.011}^{0.089} \frac{\partial \theta_s}{\partial x_2} ds. \quad (4.27)$$

where the limits 0.011 and 0.089 correspond to straight edge segment of the cavity after excluding the rounded corners which have a fillet radius of 0.003 m. As done in the previous example, mesh UM1 is discretized using elements of degree of approximation, $k = 7$ and mesh UM2 with $k = 3$ for the considered coupled CG(T_{k+1})-HDG(T_k) and CG(T_{k+1})-CG($T_{k+1}T_k$) models.

Table 4.5: Computational details for coupled CG(T_{k+1})-HDG(T_k) and CG(T_{k+1})-CG($T_{k+1}T_k$) models for GFRP problem with mesh UM1 ($k = 7$) and mesh UM2 ($k = 3$) with a time step of $0.0625/2^3$.

Model	ndof	nnz	Total CPU time for linear solver (in hours)	Number of solves	CPU Time/solve (in seconds)
CG(Q_8)-HDG(Q_7)	11 898	13 03 823	80.3 hr	1 74 449	1.657 sec
CG(Q_4)-HDG(Q_3)	23 208	13 70 509	114 hr	1 74 533	2.354 sec
CG(Q_8)-CG(Q_8Q_7)	13 519	18 95 408	146 hr	1 74 452	3.027 sec
CG(Q_4)-CG(Q_4Q_3)	22 812	17 08 665	171 hr	1 74 684	3.535 sec

Figures 4.14 and 4.15 present the results of coupled CG-HDG and CG-CG models, respectively. The temperatures in the thermocouples of coupled CG-HDG and CG-CG models from both computations are very close and they fall within their respective uncertainty intervals established from the straight corner geometries. However, the average Nusselt number in the case of both coupled CG-HDG and CG-CG models show ambiguous behaviour. One of the main reasons is the disappearance of singularities in the geometry with rounded corners. In the present case, the average Nusselt number is computed excluding the curved part of the boundary, as presented in equation (4.27). This might be the reason for the under-prediction of the average Nusselt number as it is computed for a boundary length of only 0.078 m instead of 0.084 m in the case of the geometry with straight corners. Another reason might be the lack of boundary refinement to resolve the thermal boundary layer near the internal edges of the cavity. There is also a possibility that this result has bigger range of uncertainty with it. Nevertheless, all other results of both coupled models with unstructured meshes are in agreement with the uncertainty bounds.

Table 4.5 presents the CPU times for linear solvers for the computations considered in this section. The conclusions drawn from the previous results hold in this section too. Even though there is significant difference in **ndof** between different degrees of approximations within each coupled model, the **nnz** count is very similar. Computations on high-order coarser meshes take significantly less time than the low-order finer meshes. The total number of solves of the system in both coupled models for different approximations is nearly the same.

4.6.6 Adaptive time step results

The results of temperatures in thermocouples and average Nusselt number using the adaptive time step scheme proposed in Section 2.8 are presented. The tolerance in the local error, δ_{tol} , is chosen to be 10^{-3} . The solution is started with a small time step of 10^{-2} for the first 60 sec. The adaptive time step scheme starts to adapt only after the first 60 sec. This ensures the errors due to initial conditions are not propagated in time. The results from the coupled CG(Q_2)-HDG(Q_1) and CG(Q_2)-CG(Q_2Q_1) models are presented here using mesh M4. Due to the limitation of computational resources available, only results till 250 sec are presented in this section. It is noticed that both coupled models adapt the time step to an order of 10^{-4} sec during the solution process, which demands a very large number of time steps to compute the solution for 400 sec.

Figures 4.16 and 4.17 present the variation of temperatures in the thermocouples and

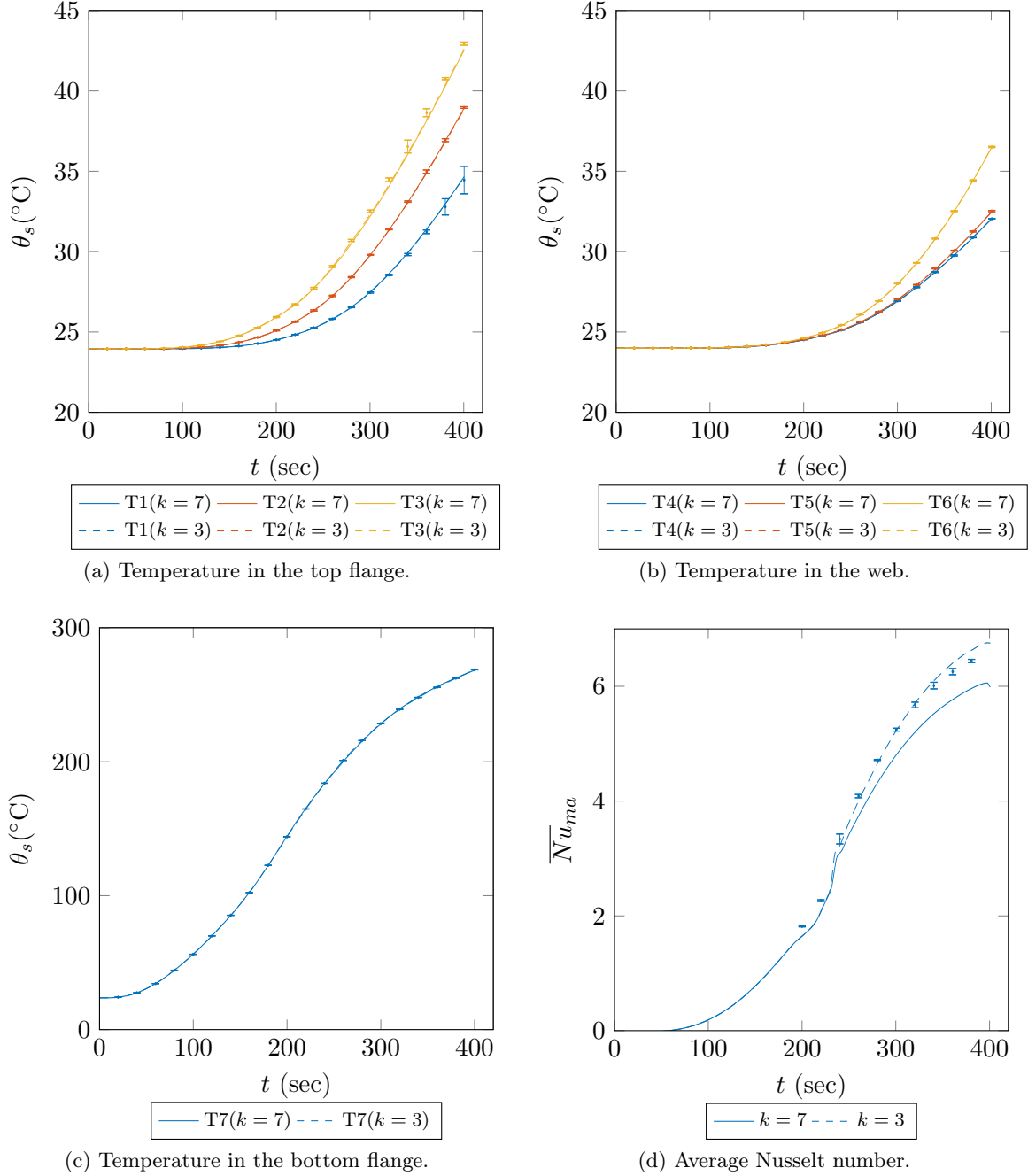


Figure 4.14: GFRP problem: evolution temperature in thermocouples and average Nusselt number with time using geometry with rounded corners and coupled CG(T_{k+1})-HDG(T_k) model. Solid lines and dashed lines correspond to UM1 and UM2 meshes, respectively and time step used is $0.0625/2^3$.

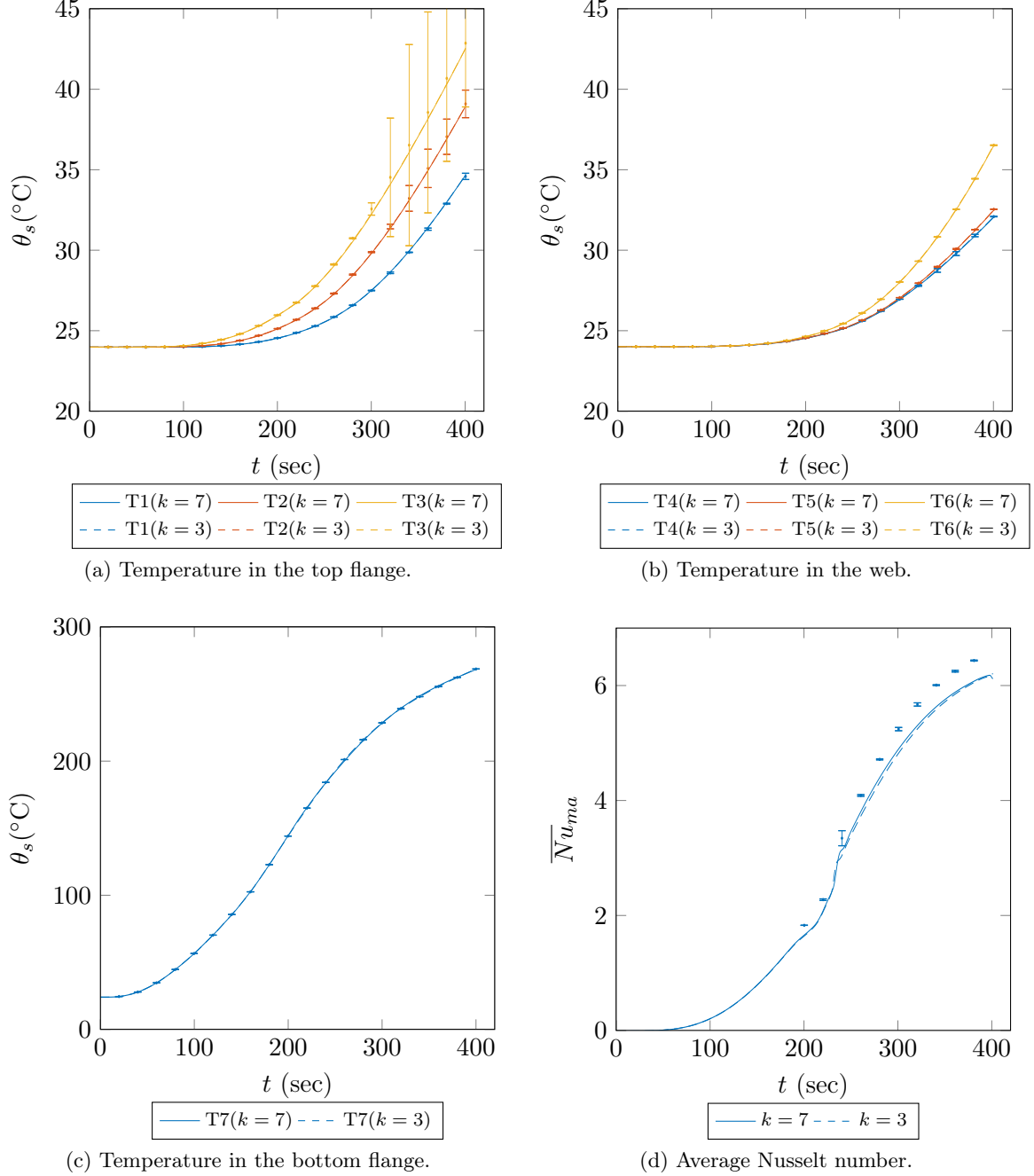


Figure 4.15: GFRP problem: evolution temperature in thermocouples and average Nusselt number with time using geometry with rounded corners and coupled $CG(T_{k+1})$ - $CG(T_{k+1}Q_k)$ model. Solid lines and dashed lines correspond to UM1 and UM2 meshes, respectively and time step used is $0.0625/2^3$.

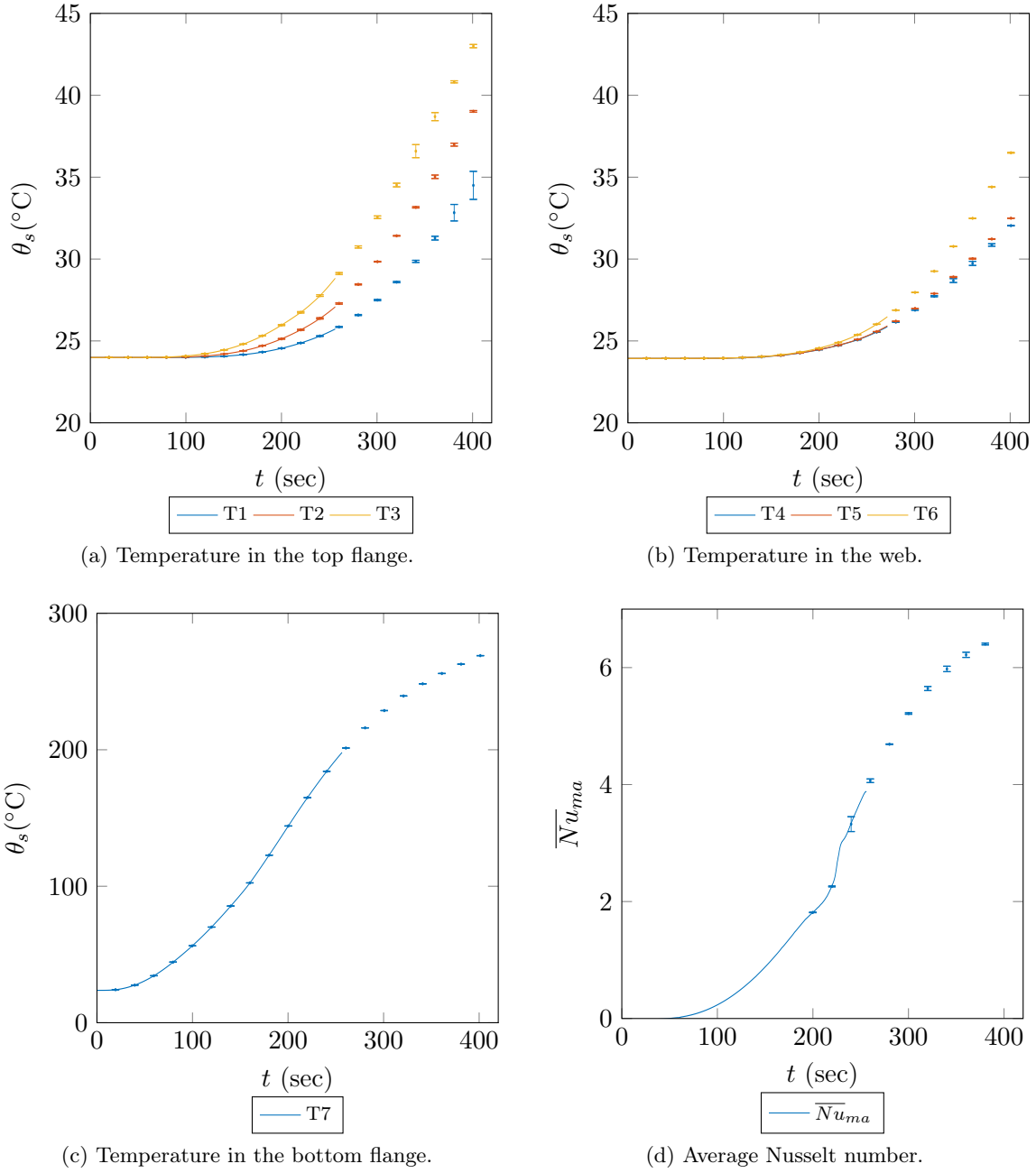
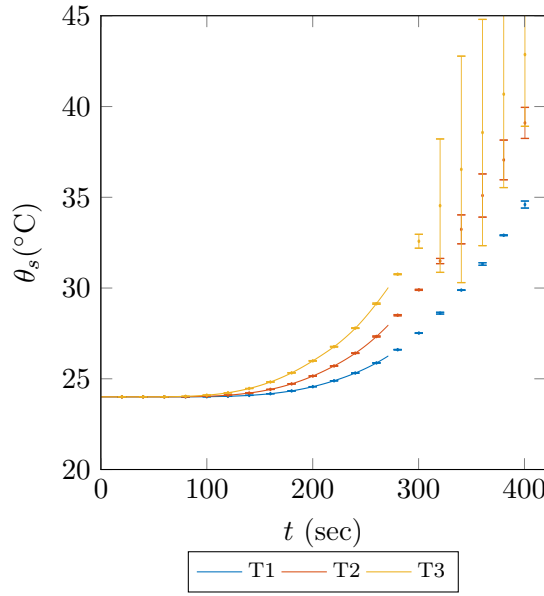


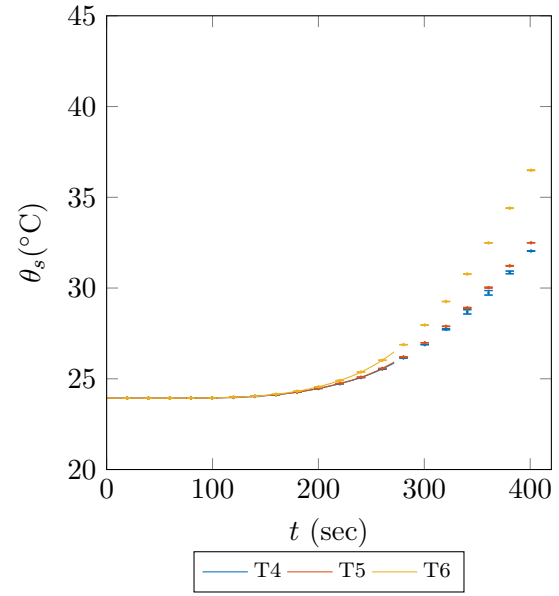
Figure 4.16: GFRP problem: evolution temperature in thermocouples and average Nusselt number with time using mesh M4, variable time step and coupled CG(Q_2)-HDG(Q_1) model.

average Nusselt number with time for coupled CG-HDG and CG-CG models, respectively. As it can be noticed from the plots the temperatures in thermocouples and average Nusselt numbers are in good agreement with the uncertainty estimates. In order to perform a consistent comparison, the moving average of Nusselt number is computed by taking an interval that spans 1 second on either side of the considered value.

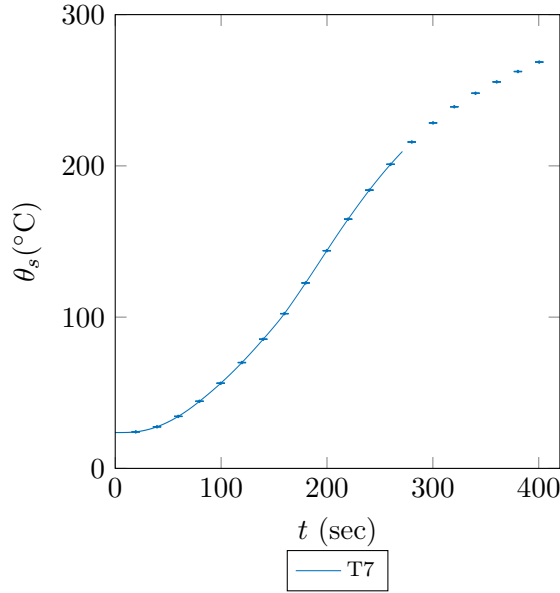
Figure 4.18a shows the comparison of adapted time step with time for both coupled CG-HDG and CG-CG models. Firstly, both coupled models adapt the time step in a similar



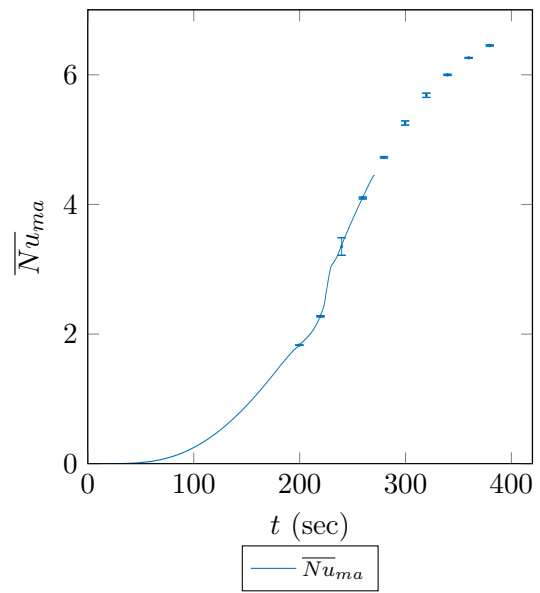
(a) Temperature in the top flange.



(b) Temperature in the web.



(c) Temperature in the bottom flange.



(d) Average Nusselt number.

Figure 4.17: GFRP problem: evolution temperature in thermocouples and average Nusselt number with time using mesh M4, variable time step and coupled $\text{CG}(Q_2)$ - $\text{CG}(Q_2Q_1)$ model.

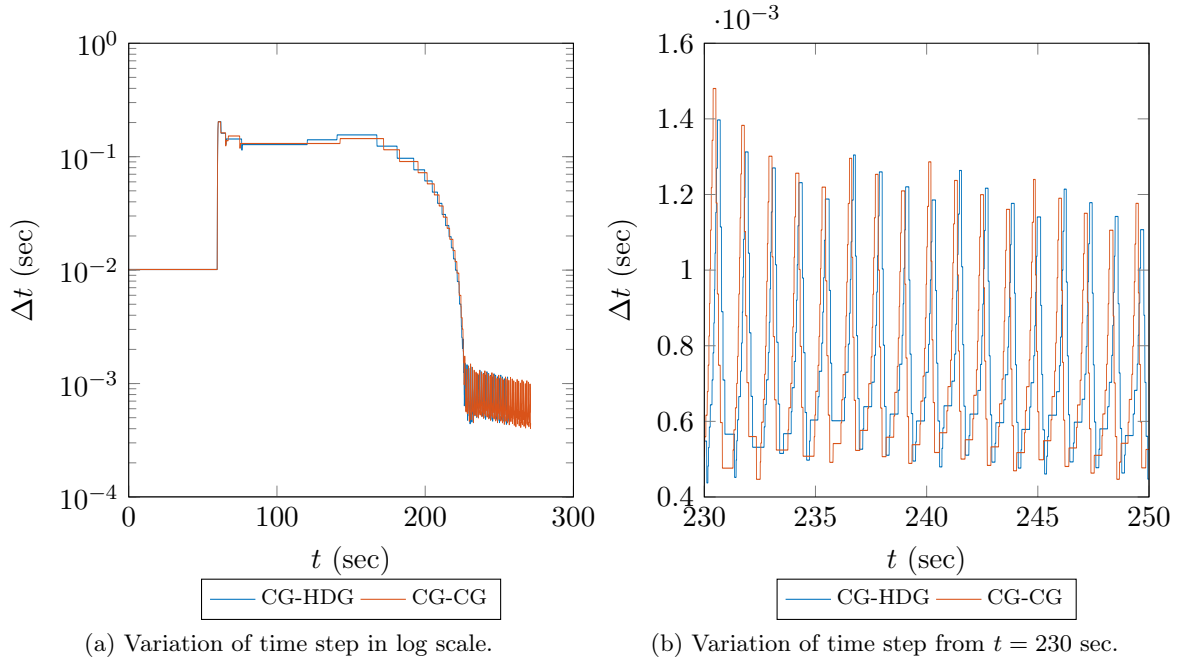


Figure 4.18: GFRP problem: variation of time step with time for the considered coupled models.

Table 4.6: Number of time steps needed for reaching different time instances for considered coupled models.

Coupled Model	Time instance, t			
	100	150	200	250
CG-HDG	6 304	6 674	7 124	43 569
CG-CG	6 298	6 683	7 142	44 769

way with the evolution of flow. This suggests that the adaptive scheme is robust and less dependent on the space discretization. It can be noticed that the time step adapts to a value of approximately 10^{-1} sec, after an initial constant time step of 10^{-2} sec. In this phase, time step almost remains constant around 10^{-1} sec until 200 sec. This is the time period where flows develops into two counter rotating symmetric vortices in the cavity. When the flow loses symmetry, which happens around 220 – 240 sec, time step starts to decrease before eventually settling into the oscillatory phase. Figure 4.18b presents the variation of time step from time $t = 230$ sec. From fig. 4.19, it can be observed that the time step changes in a periodic behaviour, which infers that the local error changes in a periodic way. The time step increases moving from a trough of the average Nusselt number to the following crest and starts decreasing after the crest. Hence, it can be inferred that the local error at the trough of the solution is more compared to the crest. As it can be noticed from plots, a similar behaviour is obtained for both coupled models in adapting the time step of the solution.

The number of time steps needed to reach different time instances for the considered coupled models are presented in table 4.6. It is clear that both models takes relatively bigger time steps till 200 sec and then, time step reduces considerably due to the instabilities in

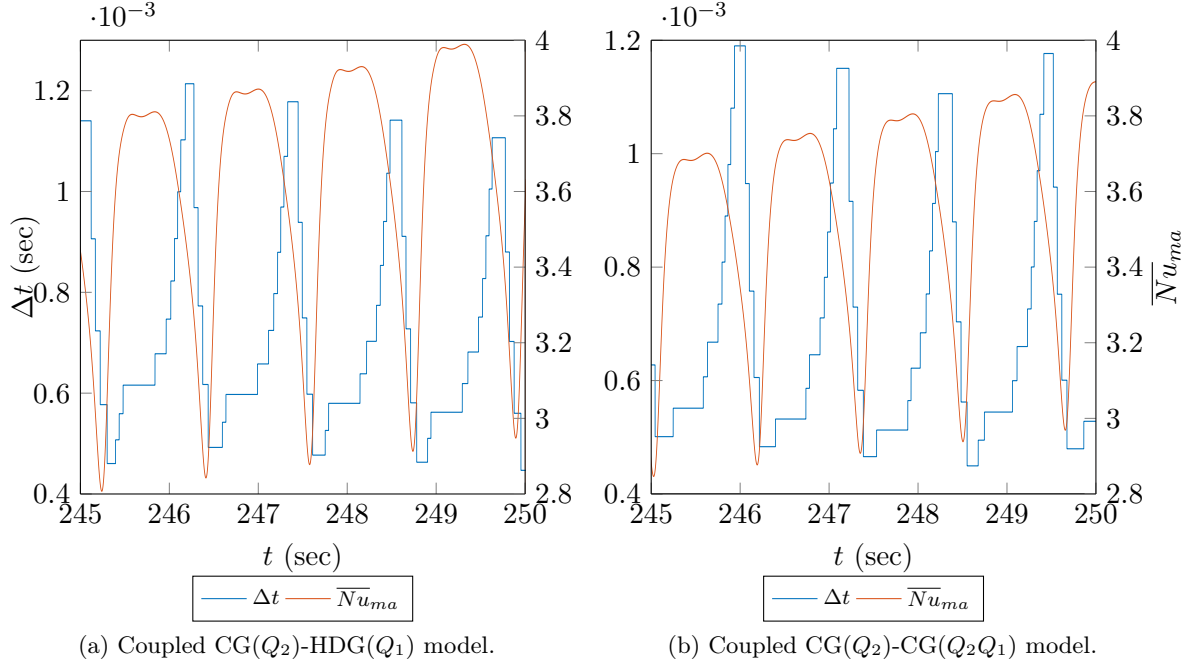


Figure 4.19: GFRP problem: variation of time step and average Nusselt number with time for the considered coupled models.

the flow. Coupled CG-HDG model takes marginally fewer time steps to reach 250 sec than coupled CG-CG model. However, both coupled models are very similar in adapting the time step, as suggested by the various results presented in the section.

4.6.7 Experimental validation

Finally, this chapter is concluded by presenting the experimental results available for this problem. Figure 4.20 shows the experimental validation of temperatures in thermocouples for the coupled CG-HDG numerical model with mesh M4 and time step of $0.0625/2^3$ sec. The experimental data is obtained from the works of Morgado *et al.* (2013b,a).

It can be noticed from the plots that the experimental results are not in agreement with the numerical ones. The discrepancy is more pronounced in the case of T5 and T6 as shown in fig. 4.20b. This might be due to the fact that experimental data has its own uncertainty. It is observed from the experimental data that the initial temperatures in different thermocouples have values between $24^\circ\text{C} - 29^\circ\text{C}$. Hence, the data has uncertainty in input parameters as well. Following Vaz *et al.* (2016), the validation uncertainty, U_{val} , can be expressed as

$$U_{val}^2 = U_{num}^2 + U_{exp}^2 + U_{inp}^2, \quad (4.28)$$

where U_{num} is the numerical uncertainty, U_{exp} is the experimental uncertainty and U_{inp} is the input parameters uncertainty. The uncertainty in the numerical results is quantified and it is concluded that it carries a relatively small error. Therefore, the difference between the experimental and numerical results can be from uncertainties in experiments and input data.

It can also be noticed that the thermocouple T6 is placed closer to the bottom face where the GFRP is exposed to fire. In the physical experiments, this is carried out in an oven by

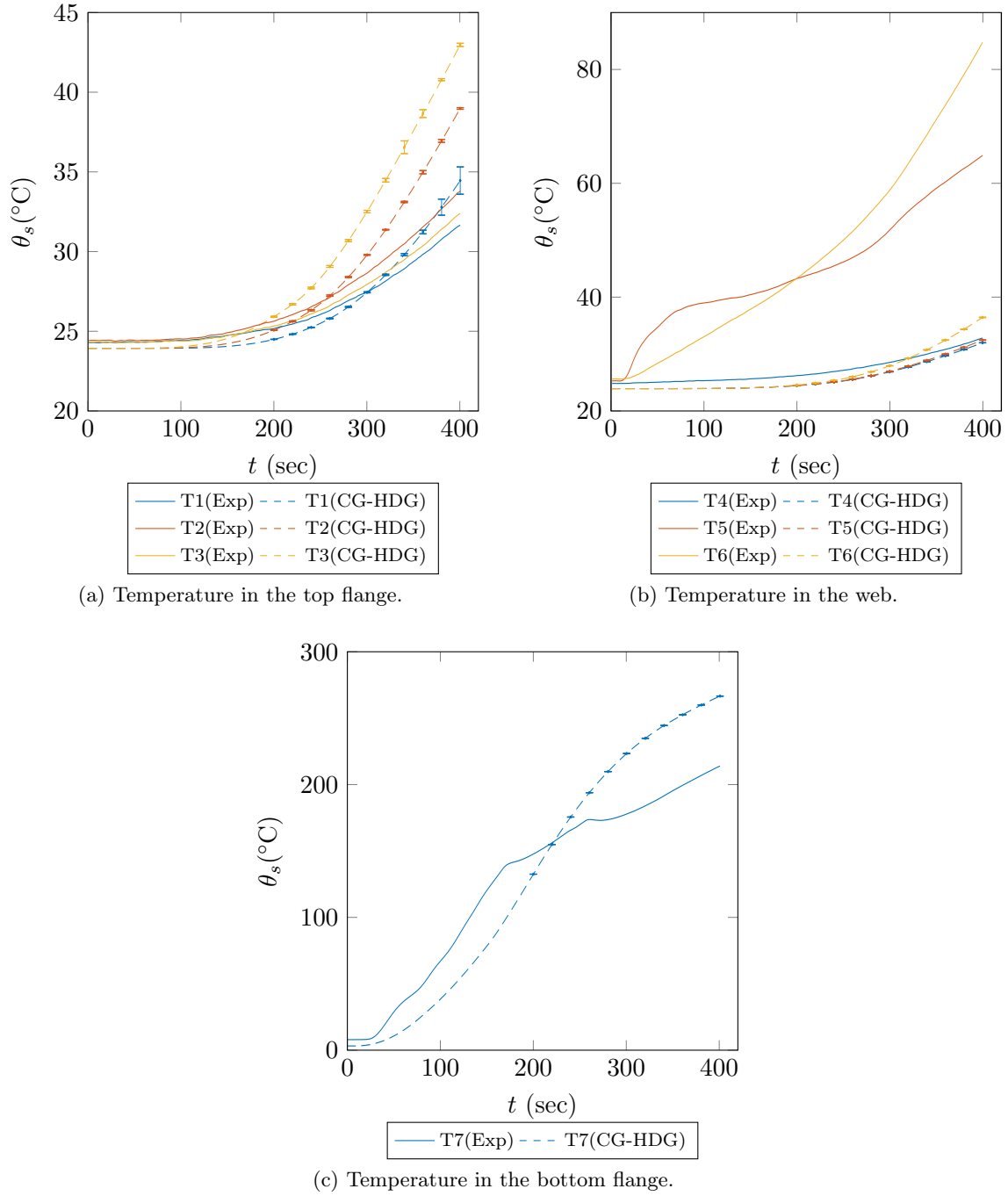


Figure 4.20: GFRP problem: experimental data of temperature in thermocouples and numerical results of coupled $CG(Q_2)$ - $CG(Q_2Q_1)$ model with time using mesh M4 and time step $0.0625/2^3$ sec.

insulating the lateral faces with wool. By inspecting the experimental results, this insulation is not well installed and hence, an abnormal rise in the temperature in T5 and T6 is observed which are closer to exposed fire. This argument is supported by thermocouple T4, which agrees well with the numerical results. The agreement of temperature in this thermocouple can be attributed to its location, which is far away from the exposed fire and hence, better insulated. Similarly, thermocouples in the top flange show an ambiguous variation experimentally. From the fig. 4.20a, it can be observed that the temperature in thermocouple T2 is higher than in T3, even though T3 is placed closer to exposed face. This can be due to congested installation of three thermocouples in a small section of 8 mm thickness of GFRP section. The temperature in the bottom flange also shows a significant difference from numerical results, however, the slopes of the two curves are in good agreement suggesting that the numerical model could capture the solution qualitatively.

4.7 Conclusions

Engineering parameters of importance, like temperature in different sections of thermocouples and average Nusselt number, are used to present the results of thermal response of GFRP tubular cross-section. Discretization errors are computed by using the results from different meshes and time steps. The computed discretization errors are used to estimate the numerical uncertainty in quantities of interest.

The uncertainty estimates from both coupled CG-HDG and CG-CG models are very similar when a good fit is obtained. The intervals either match or overlap in most of the cases for both coupled models. However, it is observed that CG-HDG model takes lesser CPU time for linear solver to obtain the solution for the total considered time. It is also noticed that the number of solves between the two coupled models is very similar. The results of high-order elements are compared with the numerical uncertainty that is computed with low-order elements. It is noticed that the results are in good agreement for the most part. In both coupled models, it is observed that using coarser high-order meshes is more efficient than finer low-order mesh. Also, coupled CG-HDG model takes lesser CPU time than the coupled CG-CG counterpart when results from same mesh are compared. Results from the geometry with rounded corners in the cavity are also presented and inferred that the temperatures are in good agreement while the average Nusselt numbers show significant differences.

Finally, the adaptive time stepping scheme results are provided for both coupled models. The temperature in the thermocouples and average Nusselt number are within the uncertainty intervals for both coupled models. The variation of time step is very similar for both coupled models. When the flow loses symmetry, the time step decreases to resolve the flow more accurately. It is also noticed that the time step goes into an oscillatory phase in both models. A comparison of number of time steps needed by both coupled models to reach 250 sec concludes that coupled CG-HDG model takes slightly lesser time steps than coupled CG-CG model. The chapter is concluded by comparing the experimental results with coupled CG-HDG numerical results for the temperatures in thermocouples of GFRP and discrepancies between numerical and experimental data are attempted to be explained.

Chapter 5

Summary and future developments

5.1 Summary

The first part of this thesis involved in devising the CG-CG formulation for the problem of a GFRP tubular cross-section. The weak formulations were derived and a code was successfully implemented and rigorously tested with various verification examples. Some of the crucial convergence results are presented in the thesis and the optimal rates are observed in all of the numerical tests. Previous works (López, 2017) on predicting the thermal behaviour of GFRP tubular cross-section problem indicated the presence of instabilities in the solution of fluid in the cavity. Hence, one of the main goals defined for this work was to discretize the cavity part of the same GFRP cross-section with a state-of-the-art HDG method. In the literature, HDG methods are claimed to be superior to conventional CG and some DG methods in terms of stability and convergence properties. Regarding computational efficiency, the work of Kirby *et al.* (2012) focused on comparison of CPU times between HDG and CG for the Laplace equation in 2-D and concluded that HDG can be as efficient as CG for high-degree approximations. However, there were no studies on the computational efficiency of HDG compared to CG for Stokes/Navier–Stokes equations were available in the literature. Hence as a preliminary study, a computational efficiency study was made between HDG and CG for Stokes/Navier–Stokes problems by implementing a HDG code on the same platform as CG. The advantage of using high-degree approximations in terms of CPU time for linear solver was demonstrated for both CG and HDG frameworks. Comparison of CPU times concluded that HDG presents a superior computational efficiency than CG for high-degree approximations. This conclusion was in-agreement with the earlier study of Kirby *et al.* (2012). In the NACA airfoil example, the ratios of CPU times for $k \geq 5$ were observed to be around 0.5 – 0.6 favouring HDG, while in the work of Kirby *et al.* (2012), the best ratio of CPU time obtained was only around 0.8 – 0.9, also favouring HDG. It is to be noted that these ratios can be dependent on the choice of linear solvers. Nonetheless, this study showed that HDG can have higher gains in computational efficiency for fluid flow problems at low Reynolds numbers when compared to CG.

Following this work, a comparative study on stability between HDG and CG for Navier–Stokes problem was made. With the aid of a manufactured solution that produces a sharp front near the boundary, it was concluded that HDG has superior stability properties than CG. No stabilization methods like SUPG were considered for CG in this study while, the stabilization parameter, τ_u , was tuned to obtain the convergence of non-linear solver in the case of HDG. It could be argued that the study is biased towards HDG because of the tuning of τ_u , which incorporates upwinding-like stabilisation. However, it should be noted

that the stabilization parameter, τ_u , is inherent to the HDG formulation, without which the convergence and uniqueness of the method cannot be proved. Hence, the stabilization constant, τ_u , can be regarded as an added parameter in addition to mesh size, h , and degree of approximation, k , that can be tuned while seeking for a solution. Stabilization techniques like SUPG requires an additional implementation overhead for CG, while in HDG this is not the case.

The second part of the thesis was to propose a coupled CG-HDG formulation for the conjugate heat transfer problem, where fluid and solid domains would be discretized by HDG and CG, respectively. The coupled variables between fluid and solid domains in this type of problem are temperature and heat flux. Hence, a coupled CG-HDG formulation was derived for the heat conduction equation. From the numerical experiments, it was understood that the CG discretization needed to be one degree higher than HDG to keep the optimal convergence in both methods. Later, an alternative coupled CG-HDG formulation was presented, which loosely based on similar idea as the first proposed model. The projection technique was used to impose the continuity of temperature and fluxes at the interface in this alternative coupled model. From the convergence tests, it was concluded that both coupled models provided similar results. The previous HDG formulation that was applied to the Navier–Stokes equations was then extended to the coupled Navier–Stokes/convection-diffusion model and the formulation was verified using convergence tests. Results of Rayleigh–Bénard convection flow are presented and results of CG and HDG methods were compared. Finally, the proposed coupled CG-HDG model was integrated with the coupled Navier–Stokes/convection-diffusion equations to solve conjugate heat transfer problems. A comparison of CPU times for linear solver for the coupled CG-HDG and CG-CG models was provided and it was observed that CG-HDG was more efficient at low and high-degree approximations. Besides, a theoretical count of number of non-zero entries was derived for both coupled CG-HDG and CG-CG models. In order to make this study a complete one, a coupled HDG-HDG model was formulated and implemented for the conjugate heat transfer problem. It was noticed that the accuracy and efficiency of HDG-HDG model, in the examples considered in this work, was very similar to proposed CG-HDG coupled model.

The final part of the thesis presents the application of the developed CG-HDG formulation to the prediction of the thermal behaviour of GFRP tubular cross-section. In this problem, the internal radiation within the cavity should be taken into account in the conjugate heat transfer model. Hence, the discretization of radiosity equation, which governs internal radiation, was detailed using the Galerkin formulation. The internal radiation model was added to the coupled CG-HDG conjugate heat transfer model. Since, this problem do not have an analytical solution, *a priori* error estimation was not possible. Therefore, *a posteriori* error estimation methods were considered to present the results. Computations with different meshes in space and time were considered to estimate the discretization error. This problem demands a very small time step and finer meshes to be in the asymptotic range of the respective discretizations. This requires more computational resources and advanced implementations (like parallelization) which is outside the scope of this thesis. As non-monotonic convergence was noticed in results, the estimated discretization errors were transformed into uncertainty to establish confidence intervals. Again, the coupled CG-HDG model outperforms the coupled CG-CG model in terms of computational efficiency.

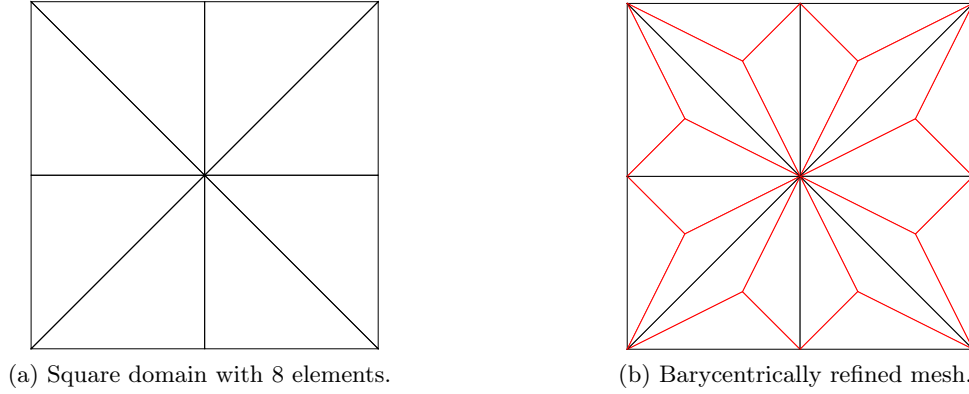


Figure 5.1: Barycentric refinement for Scott–Vogelius elements.

5.2 Future work

5.2.1 Comparison between CG and HDG

There are plenty of other aspects can be considered in the comparison between CG and HDG for incompressible Navier–Stokes equations. One of the important being the usage of iterative solvers to solve the linear system of equations. In very large test cases in 2-D and most of the applications in 3-D limits the usage of direct solvers because of large memory requirements. Hence, a comparative study between two methods with iterative solvers can be of significant importance to understand the relative merits of both discretizations. The work of Yakovlev *et al.* (2016) concluded that there is need to develop effective preconditioners for HDG to solve the linear systems arising from Laplace equation in 3-D. Extending this conclusion to linear systems arising from Stokes/Navier–Stokes equations is not straightforward as they result in saddle point systems, unlike the positive definite matrices arising in the Laplace equation problem.

In this work only Taylor–Hood elements were considered for CG discretization of Navier–Stokes equations. However, there has been recent developments in divergence-free elements like Scott–Vogelius (Scott and Vogelius, 1985). This type of elements use discontinuous approximations for pressure on barycentrically-refined meshes as shown in fig. 5.1. Hence, all the pressure DOFs have local support which can be reduced to a single unknown per each element in the global system by static condensation. Cousins *et al.* (2013) did a comparative study between Scott–Vogelius and Taylor–Hood elements. The work demonstrated that Scott–Vogelius have potential gains in efficiency of iterative solvers in 3-D over Taylor–Hood elements. Besides, divergence-free velocity field satisfies the mass conservation equation in Scott–Vogelius elements more accurately. The importance of the mass conservation property in the context of buoyancy flows is demonstrated in the work of Keith *et al.* (2012). However, the apparent restrictions on the meshes of Scott–Vogelius elements limit their application to practical problems, like the one considered in the present work. Nevertheless, it would be interesting to compare the accuracy and efficiency of Scott–Vogelius elements with divergence-free HDG formulations for Navier–Stokes equations.

As stated in Chapter 1, Embedded Discontinuous Galerkin (EDG) (Güzey *et al.*, 2007) and Multi-scale Discontinuous Galerkin (MDG) (Hughes *et al.*, 2006) methods use continuous trace approximation which results in identical number of DOFs for velocity compared to CG.

Hence, they result in smaller number of overall DOFs compared to HDG. Even though they do not possess the super-convergence property, adding the results from EDG and MDG would improve the quality of the comparison.

5.2.2 Coupled CG-HDG formulation

At the start of this work, it was assumed that using CG in the solid part of conjugate heat transfer problem is the best choice in terms of computational efficiency. The rationale behind that choice was that the solution in solid part is smooth, which does not need stabilisation and also, CG has lower number of DOFs compared to HDG for second-order elliptic operators. Hence, the present coupled CG-HDG formulation was devised for heat equation. However, during the course of the work, it was inferred that the coupled HDG-HDG formulation is very similar in terms of CPU time for linear solver and memory requirements. These conclusions hold for the examples considered in the present work using direct solvers. The computational efficiency and complexity of the proposed algorithm can be studied using problems with bigger computational domains and iterative solvers.

In addition, the proposed coupled CG-HDG formulation can be extended to incompressible Navier–Stokes equations. HDG can be used in the part of the domain where discontinuities and/or boundary layers are present and CG in the part where solution is smooth. This coupled formulation enhances the computational efficiency at low-order approximations of HDG at the same time retaining the superior stability properties. Finally, the mathematical analysis of the proposed coupled formulation for second-order elliptic operators needs to be developed. In the present work, convergence is established using numerical experiments. However, there is a scope to develop the mathematical analysis of existence and uniqueness of the coupled formulation.

5.2.3 High-order time integration

The BDF schemes considered in this work are not A-stable for order, $\mu > 2$. Hence, high-order BDF schemes are not recommended. Several choices for high-order time integration are available like Runge–Kutta (RK) methods (Hairer *et al.*, 1993), which are unconditionally stable. Let $\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y})$ be an initial value problem with initial condition $\mathbf{y}(t_0) = \mathbf{y}_0$, then a s –stage RK method can be expressed as follows,

$$\begin{aligned} \mathbf{y}_{n+1} &= \mathbf{y}_n + \Delta t_n \sum_{i=1}^s b_i \mathbf{k}_i, \\ \mathbf{k}_i &= \mathbf{f}(t_n + c_i \Delta t_n, \mathbf{u}_n + \Delta t_n \sum_{j=1}^s a_{ij} \mathbf{k}_j), \quad i = 1, \dots, s, \end{aligned} \quad (5.1)$$

where coefficients in equation (5.1) are usually represented by Butcher table,

$$\begin{array}{c|ccc} c_1 & a_{11} & \dots & a_{1s} \\ c_1 & a_{11} & \dots & a_{1s} \\ \vdots & \vdots & \vdots & \vdots \\ c_s & a_{s1} & \dots & a_{ss} \\ \hline & b_1 & \dots & b_s \end{array} = \frac{\mathbf{c}}{\mathbf{b}^T} \bigg| \begin{array}{c} \mathbf{A} \\ \mathbf{b}^T \end{array}. \quad (5.2)$$

If ndof is the total number of DOFs of a system, a s –stage fully implicit Runge–Kutta (IRK) scheme requires to solve a non-linear system of dimension $s \text{ndof}$. Hence, IRK schemes are rather computationally expensive. On the other hand, s –stage Diagonally Implicit Runge–Kutta (DIRK) methods are devised by keeping $a_{ij} = 0$ for $i < j$ which require solutions

to s non-linear system of equations with dimension ndof . However, DIRK methods suffers from order reduction for Differential Algebraic Equation (DAE) systems (Hairer *et al.*, 1993, Rang, 2007). The pressure can only be approximated with first or second-order using DIRK methods (Montlaur *et al.*, 2012, Rang, 2016).

A linearisation of DIRK method leads to so-called Rosenbrock–Wanner (ROW) method which can be expressed as follows,

$$\mathbf{k}_i = \mathbf{f}(t_n + \alpha_i \Delta t_n, \mathbf{u}_n + \Delta t_n \sum_{j=1}^{i-1} \alpha_{ij} \mathbf{k}_j) + \Delta t_n \mathbf{f}_y \sum_{j=1}^i \gamma_{ij} \mathbf{k}_j + \Delta t_n \gamma_i \mathbf{f}_t(t_n, \mathbf{y}_n), \quad i = 1, \dots, s, \quad (5.3)$$

where \mathbf{f}_y and \mathbf{f}_t denotes the Jacobian and time derivative of \mathbf{f} in equation (5.3). This scheme requires only the solution of s linear systems, when s – stage ROW method is employed at each time step. This is due to the fact that left hand side matrix is independent of the stage number. This scheme can be computationally efficient as it not necessary to solve a non-linear system of equations at each time, but only fixed number of systems. Nevertheless, this scheme has the limitation of order reduction too. There are different ROW methods proposed in the literature that can attain orders of 3 and 4 (Lubich and Roche, 1990, Steinebach, 1995, Rang and Angermann, 2005). Moreover, in the work of Montlaur *et al.* (2012), the author concluded that ROW method can be computationally efficient at low to moderate levels of accuracy. Hence, ROW method of time integration scheme can be beneficial in the context of present work as it practically requires only one factorisation of linear system per each time step. In addition, embedded schemes can be devised for ROW and Runge-Kutta time integration schemes to compute the local error that can be used to adapt the time step without additional computational overhead.

5.2.4 Code development

The present code was implemented in FORTRAN using open-source libraries to solve the linear system of equations. As presented in Chapter 4, the solution of the considered GFRP tubular cross-section required very fine meshes with small time steps to predict the thermal behaviour. Even though several optimizations were used in the code, there is still a potential to further improve the code. In the present work, all the elemental matrices are pre-computed including the non-linear ones. This ensures that the loop in Gauss points is eliminated at each non-linear iteration to assemble the local elemental matrices. A big improvement on total CPU time was noticed, especially at high-order approximations with this implementation. However, there is still a considerable amount of CPU time spent on condensing the elemental matrices. This cost is more pronounced in the case of HDG, as the local unknowns are more than CG ones. One way to optimize this part to the code is to use shared memory computing, *i.e.*, OpenMP. This is relatively easy to include in the code and compute the elemental matrices in parallel on several threads. More complicated parallelization algorithms can also be considered like domain decomposition methods (Li *et al.*, 2014). However, in the context of the present work, it is not strictly necessary. Regarding the linear solvers, multi-thread supported like PARDISO (Petra *et al.*, 2014b,a) linear solvers can also be used to improve the total run time.

Appendix A

Weak formulations for coupled CG-CG and HDG-HDG models

The weak formulation of the coupled HDG(T_k/Q_k)-HDG(T_k/Q_k) model is presented for the conjugate heat transfer problem. Using the notation, $\mathbf{L} \approx \text{grad } \mathbf{u}$, $\mathbf{q}_1 \approx -\alpha_1 \text{grad } \theta_1$ and $\mathbf{q}_2 \approx -\alpha_2 \text{grad } \theta_2$, the weak form corresponding to the equations (3.41) can be stated as follows: find $(\mathbf{u}, p, \mathbf{L}, \theta_1, \mathbf{q}_1, \theta_2, \mathbf{q}_2, \hat{\mathbf{u}}, \rho_e, \hat{\theta}_1, \hat{\theta}_2) \in [\mathcal{V}_k^h(\Omega_1)]^2 \times \mathcal{V}_k^h(\Omega_1) \times [\mathcal{V}_k^h(\Omega_1)]^{2 \times 2} \times \mathcal{V}_k^h(\Omega_1) \times [\mathcal{V}_k^h(\Omega_1)]^2 \times \mathcal{V}_k^h(\Omega_2) \times [\mathcal{V}_k^h(\Omega_2)]^2 \times [\Lambda_k^h(\Gamma_1)]^2 \times \mathbb{R}^{m_{el}} \times \Lambda_k^h(\Gamma_1) \times \Lambda_k^h(\Gamma_2)$ satisfying the *local problem* in every element Ω_1^e and Ω_2^i ,

$$\begin{aligned} & (\delta \mathbf{L}, \mathbf{L})_{\Omega_1^e} + (\text{div } \delta \mathbf{L}, \mathbf{u})_{\Omega_1^e} - \langle \delta \mathbf{L} \mathbf{n}, \hat{\mathbf{u}} \rangle_{\partial \Omega_1^e} = 0, \\ & - (\text{grad } \delta \mathbf{u}, \mathbf{u} \otimes \mathbf{u})_{\Omega_1^e} - (\delta \mathbf{u}, \text{div } (-p \mathbf{I} + \nu \mathbf{L}))_{\Omega_1^e} \\ & \quad + \langle \delta \mathbf{u}, (\hat{\mathbf{u}} \otimes \hat{\mathbf{u}}) \mathbf{n} + \tau_u (\mathbf{u} - \hat{\mathbf{u}}) \rangle_{\partial \Omega_1^e} - (\delta \mathbf{u}, \mathbf{f}(\theta_1))_{\Omega_1^e} - (\delta \mathbf{u}, \bar{\mathbf{s}})_{\Omega_1^e} = 0, \end{aligned} \quad (\text{A.1a})$$

$$\begin{aligned} & - (\text{grad } \delta p, \mathbf{u})_{\Omega_1^e} + \langle \delta p, \hat{\mathbf{u}} \cdot \mathbf{n} \rangle_{\partial \Omega_1^e} = 0, \\ & \frac{1}{|\partial \Omega_1^e|} \langle p, 1 \rangle_{\partial \Omega_1^e} = \rho_e, \end{aligned} \quad (\text{A.1b})$$

$$\begin{aligned} & (\delta \mathbf{q}_1, \alpha_1^{-1} \mathbf{q}_1)_{\Omega_1^e} - (\text{div } \delta \mathbf{q}_1, \theta_1)_{\Omega_1^e} + \langle \delta \mathbf{q}_1 \cdot \mathbf{n}, \hat{\theta}_1 \rangle_{\partial \Omega_1^e} = 0, \\ & (\delta \theta_1, \text{div } \mathbf{q}_1)_{\Omega_1^e} - (\text{grad } \delta \theta_1, \mathbf{u} \theta_1)_{\Omega_1^e} + \langle \delta \theta_1, (\hat{\mathbf{u}} \cdot \mathbf{n} - \tau_\theta) \hat{\theta}_1 \rangle_{\partial \Omega_1^e} \\ & \quad + \langle \delta \theta_1, \tau_\theta \theta_1 \rangle_{\partial \Omega_1^e} - (\delta \theta_1, \bar{g}_1)_{\Omega_1^e} = 0, \end{aligned} \quad (\text{A.1c})$$

$$\begin{aligned} & (\delta \mathbf{q}_2, \alpha_2^{-1} \mathbf{q}_2)_{\Omega_2^i} - (\text{div } \delta \mathbf{q}_2, \theta_2)_{\Omega_2^i} + \langle \delta \mathbf{q}_2 \cdot \mathbf{n}, \hat{\theta}_2 \rangle_{\partial \Omega_2^i} = 0, \\ & (\delta \theta_2, \text{div } \mathbf{q}_2)_{\Omega_2^i} + \langle \delta \theta_2, \tau_\theta (\theta_2 - \hat{\theta}_2) \rangle_{\partial \Omega_2^i} - (\delta \theta_2, \bar{g}_2)_{\Omega_2^i} = 0, \end{aligned} \quad (\text{A.1d})$$

for $e = 1, \dots, m_{el}$, and $i = 1, \dots, p_{el}$, and the *global problem*

$$\begin{aligned} & \sum_{e=1}^{m_{el}} \langle \delta \hat{\mathbf{u}}, (-p \mathbf{I} + \nu \mathbf{L}) \mathbf{n} + \tau_u (\hat{\mathbf{u}} - \mathbf{u}) \rangle_{\partial \Omega_1^e} = 0, \\ & \langle \hat{\mathbf{u}} \cdot \mathbf{n}, 1 \rangle_{\partial \Omega_1^e} = 0 \quad \text{for } e = 1, \dots, m_{el}, \\ & \hat{\mathbf{u}} = \mathbb{P}_2(\bar{\mathbf{u}}) \quad \text{on } \partial \Omega_1, \end{aligned} \quad (\text{A.2a})$$

$$\begin{aligned} \sum_{e=1}^{m_{el}} \langle \delta \hat{\theta}_1, (\mathbf{q}_1 + \hat{\mathbf{u}} \hat{\theta}_1) \cdot \mathbf{n} + \tau_\theta(\theta_1 - \hat{\theta}_1) \rangle_{\partial \Omega_1^e} + \sum_{i=1}^{p_{el}} \langle \delta \hat{\theta}_2, \mathbf{q}_2 \cdot \mathbf{n} + \tau_\theta(\theta_2 - \hat{\theta}_2) \rangle_{\partial \Omega_2^i} \\ = \langle \delta \hat{\theta}_1, \bar{q}_{n1} \rangle_{\Gamma_1^N} + \langle \delta \hat{\theta}_2, \bar{q}_{n2} \rangle_{\Gamma_2^N}, \end{aligned} \quad (\text{A.2b})$$

$$\begin{aligned} \hat{\theta}_1 &= \mathbb{P}_2(\bar{\theta}_1) \quad \text{on } \Gamma_1^D, \\ \hat{\theta}_2 &= \mathbb{P}_2(\bar{\theta}_2) \quad \text{on } \Gamma_2^D, \end{aligned}$$

for all $(\delta \mathbf{u}, \delta p, \delta \mathbf{L}, \delta \theta_1, \delta \mathbf{q}_1, \delta \theta_2, \delta \mathbf{q}_2, \delta \hat{\mathbf{u}}, \delta \hat{\theta}_1, \delta \hat{\theta}_2) \in [\mathcal{V}_k^h(\Omega_1)]^2 \times \mathcal{V}_k^h(\Omega_1) \times [\mathcal{V}_k^h(\Omega_1)]^{2 \times 2} \times \mathcal{V}_k^h(\Omega_1) \times [\mathcal{V}_k^h(\Omega_1)]^2 \times \mathcal{V}_k^h(\Omega_2) \times [\mathcal{V}_k^h(\Omega_2)]^2 \times [\Lambda_k^h(\Gamma_1)]^2 \times \Lambda_k^h(\Gamma_1) \times \Lambda_k^h(\Gamma_2)$ such that $\delta \hat{\mathbf{u}} = \mathbf{0}$, $\delta \hat{\theta}_1 = 0$ on Γ_1^D and $\delta \hat{\theta}_2 = 0$ on Γ_2^D . Recall, Γ_2 is the union of all edges in the partitioned Ω_2 and p_{el} is the number of elements in Ω_2 .

The weak formulation for the coupled CG($T_{k+1}T_k/Q_{k+1}Q_k$)-CG(T_{k+1}/Q_{k+1}) model corresponding to the governing equations of the conjugate heat transfer problem presented in (3.41) can be stated as: find $(\mathbf{u}, p, \theta_1, \theta_2) \in [\mathcal{W}_{k+1}^h(\Omega_1)]^2 \times \mathcal{V}_k^h(\Omega_1) \times \mathcal{W}_{k+1}^h(\Omega_1) \times \mathcal{W}_{k+1}^h(\Omega_2)$ such that $\mathbf{u} = \Pi^h(\bar{\mathbf{u}})$ on $\partial \Omega_1$, $\theta_1 = \Pi^h(\bar{\theta}_1)$ on Γ_1^D , $\theta_2 = \Pi^h(\bar{\theta}_2)$ on Γ_2^D and

$$\begin{aligned} (\delta \mathbf{u}, (\text{grad } \mathbf{u}) \mathbf{u})_{\Omega_1} + (\text{grad } \delta \mathbf{u}, (-p \mathbf{I} + \nu \text{grad } \mathbf{u}))_{\Omega_1} - (\delta \mathbf{u}, \bar{\mathbf{f}}(\theta_1))_{\Omega_1} \\ - (\delta \mathbf{u}, \bar{\mathbf{s}})_{\Omega_1} = 0, \end{aligned} \quad (\text{A.3a})$$

$$\begin{aligned} - (\delta p, \text{div } \mathbf{u})_{\Omega_1} = 0, \\ (\text{grad } \delta \theta_1, \alpha_1 \text{grad } \theta_1)_{\Omega_1} + (\delta \theta_1, \mathbf{u} \cdot \text{grad } \theta_1)_{\Omega_1} - (\delta \theta_1, \bar{g}_1)_{\Omega_1} - \langle \delta \theta_1, \bar{q}_{n1} \rangle_{\Gamma_1^N} \\ + (\text{grad } \delta \theta_2, \alpha_2 \text{grad } \theta_2)_{\Omega_2} - (\delta \theta_2, \bar{g}_2)_{\Omega_2} - \langle \delta \theta_2, \bar{q}_{n2} \rangle_{\Gamma_2^N} = 0, \end{aligned} \quad (\text{A.3b})$$

for all $(\delta \mathbf{u}, \delta p, \delta \theta_1, \delta \theta_2) \in [\mathcal{W}_{k+1}^h(\Omega_1)]^2 \times \mathcal{V}_k^h(\Omega_1) \times \mathcal{W}_{k+1}^h(\Omega_1) \times \mathcal{W}_{k+1}^h(\Omega_2)$ such that $\delta \mathbf{u} = \mathbf{0}$ on $\partial \Omega_1$, $\delta \theta_1 = 0$ on Γ_1^D and $\delta \theta_2 = 0$ on Γ_2^D , where the discrete spaces are defined in (1.2).

The weak formulation for the coupled CG($T_{k+1}T_k/Q_{k+1}Q_k$)-CG(T_{k+1}/Q_{k+1}) model corresponding to the governing equations in (4.1) and (4.5) can be presented as: find $(\mathbf{u}^{n+1}, p^{n+1}, \theta_f^{n+1}, \theta_s^{n+1}, R^{n+1}) \in [\mathcal{W}_{k+1}^h(\Omega_f)]^2 \times \mathcal{V}_k^h(\Omega_f) \times \mathcal{W}_{k+1}^h(\Omega_f) \times \mathcal{W}_{k+1}^h(\Omega_s) \times \Sigma_{k+1}^h(\Gamma_R)$ such that $\mathbf{u} = \mathbf{0}$ on Γ_R and

$$\begin{aligned} \left(\delta \mathbf{u}, \frac{\alpha_\mu \mathbf{u}^{n+1}}{\Delta t} \right)_{\Omega_f} + (\delta \mathbf{u}, (\text{grad } \mathbf{u}^{n+1}) \mathbf{u}^{n+1})_{\Omega_f} + (\text{grad } \delta \mathbf{u}, (-p^{n+1} \mathbf{I} + \nu \text{grad } \mathbf{u}^{n+1}))_{\Omega_f} \\ - (\delta \mathbf{u}, \bar{\mathbf{f}}(\theta_f^{n+1}))_{\Omega_f} - \left(\delta \mathbf{u}, \frac{\mathbf{u}^{n, \text{BDF}\mu}}{\Delta t} \right)_{\Omega_f} = 0, \\ - (\delta p, \text{div } \mathbf{u}^{n+1})_{\Omega_f} = 0, \end{aligned} \quad (\text{A.4a})$$

$$\begin{aligned}
& \left(\delta\theta_f, \rho_f c_{pf} \frac{\alpha_\mu \theta_f^{n+1}}{\Delta t^{n+1}} \right)_{\Omega_f} + \left(\text{grad } \delta\theta_f, \kappa_f \text{grad } \theta_f^{n+1} \right)_{\Omega_f} + \left(\delta\theta_f, \rho_f c_{pf} (\mathbf{u}^{n+1} \cdot \text{grad } \theta_f^{n+1}) \right) \\
& + \left(\delta\theta_s, \rho_s (\theta_s^{n+1}) c_{ps} (\theta_s^{n+1}) \frac{\alpha_\mu \theta_s^{n+1}}{\Delta t^{n+1}} \right)_{\Omega_s} + \left(\text{grad } \delta\theta_s, \kappa_s (\theta_s^{n+1}) \text{grad } \theta_s^{n+1} \right)_{\Omega_s} \\
& - \left(\delta\theta_s, \rho_s (\theta_s^{n+1}) c_{ps} (\theta_s^{n+1}) \frac{\theta_s^{n, \text{BDF}\mu}}{\Delta t^{n+1}} \right)_{\Omega_s} - \left(\delta\theta_f, \rho_f c_{pf} \frac{\theta_f^{n, \text{BDF}\mu}}{\Delta t^{n+1}} \right)_{\Omega_f} \\
& + \left\langle \delta\theta_s, \frac{\epsilon}{1-\epsilon} (\sigma (\theta_s^{n+1})^4 - R^{n+1}) \right\rangle_{\Gamma_R} - \langle \delta\theta_s, h_a (\theta_s^{n+1}) (\theta_a^{n+1} - \theta_s^{n+1}) \rangle_{\Gamma_t \cup \Gamma_b} \\
& - \langle \delta\theta_s, \epsilon \sigma ((\theta_a^{n+1})^4 - (\theta_s^{n+1})^4) \rangle_{\Gamma_t \cup \Gamma_b} = 0,
\end{aligned} \tag{A.4b}$$

$$\begin{aligned}
& \sum_{e=1}^{n_l} \left\langle \delta R, \frac{R^{n+1}}{(1-\epsilon)} \right\rangle_{\Gamma_R^e} - \sum_{e=1}^{n_l} \left\langle \delta R, \sigma \frac{\epsilon}{(1-\epsilon)} (\theta_s^{n+1})^4 \right\rangle_{\Gamma_R^e} \\
& - \sum_{e=1}^{n_l} \sum_{\substack{k=1 \\ k \neq e}}^{n_l} \left\langle \delta R, \left\langle R^{n+1, k}, \frac{\cos \gamma^e \cos \gamma^k}{2r} \right\rangle_{\Gamma_R^k} \right\rangle_{\Gamma_R^e} = 0,
\end{aligned} \tag{A.4c}$$

for all $(\delta \mathbf{u}, \delta p, \delta \theta_f, \delta \theta_s, \delta R) \in [\mathcal{W}_{k+1}^h(\Omega_f)]^2 \times \mathcal{V}_k^h(\Omega_f) \times \mathcal{W}_{k+1}^h(\Omega_f) \times \mathcal{W}_{k+1}^h(\Omega_s) \times \Sigma_{k+1}^h(\Gamma_R)$ such that $\delta \mathbf{u} = \mathbf{0}$ on Γ_R .

Appendix B

Theoretical count of degrees of freedoms and number of non-zeros

All the hypothesis and notation presented in this appendix are taken from the work of Huerta *et al.* (2013). In their work, second-order elliptic equation is considered to derive the quantities of interest. As the present work deals with a coupled system of equations, the counting technique of Huerta *et al.* (2013) is applied to each block of unknowns in the system. Consider conjugate heat transfer problem that is governed by equations presented in (3.41). The fill of the global tangent matrix and its corresponding DOFs have the following pattern,

$$\begin{bmatrix}
 \begin{matrix} \ddots & & \\ & u_1 u_1 & \\ & & \ddots \end{matrix} &
 \begin{matrix} \ddots & & \\ & u_1 u_2 & \\ & & \ddots \end{matrix} &
 \begin{matrix} \ddots \\ u_1 p \\ \ddots \end{matrix} &
 \begin{matrix} \ddots & & \\ & u_1 \theta_f & \\ & & \ddots \end{matrix} &
 \\
 \begin{matrix} \ddots & & \\ & u_2 u_1 & \\ & & \ddots \end{matrix} &
 \begin{matrix} \ddots & & \\ & u_2 u_2 & \\ & & \ddots \end{matrix} &
 \begin{matrix} \ddots \\ u_2 p \\ \ddots \end{matrix} &
 \begin{matrix} \ddots & & \\ & u_2 \theta_f & \\ & & \ddots \end{matrix} &
 \\
 \begin{matrix} \ddots \\ u_1 p \\ \ddots \end{matrix} &
 \begin{matrix} \ddots \\ u_2 p \\ \ddots \end{matrix} &
 \begin{matrix} pp \\ \ddots \end{matrix} &
 &
 \\
 \begin{matrix} \ddots & & \\ & u_1 \theta_f & \\ & & \ddots \end{matrix} &
 \begin{matrix} \ddots & & \\ & u_2 \theta_f & \\ & & \ddots \end{matrix} &
 &
 \begin{matrix} \ddots & & \\ & \theta_f \theta_f & \\ & & \ddots \end{matrix} &
 \\
 & & & &
 \begin{matrix} \ddots & & \\ & \theta_s \theta_s & \\ & & \ddots \end{matrix} &
 \end{bmatrix}
 \begin{Bmatrix}
 \vdots \\
 u_1 \\
 \vdots \\
 u_2 \\
 \vdots \\
 p \\
 \vdots \\
 \theta_f \\
 \vdots \\
 \theta_s \\
 \vdots
 \end{Bmatrix} \quad (B.1)$$

In the fill-in representation (B.1), θ_f and θ_s corresponds to the DOFs associated to temperature in fluid and solid domains, respectively. Here, the temperature DOFs on the interface is not considered as the boundary nodes on interface are negligible compared to interior nodes in the domains. The global tangent matrix from the CG discretization of Stokes/Navier–Stokes equations presents a saddle point system with zeros in main diagonal. However, when static condensation is applied on high degree approximations ($k \geq 3$), the block pp represented

in (B.1) becomes non-zero. This fill-in pattern is valid for HDG discretization as well with the exception of null pp block.

First, expressions for \mathbf{ndof} for incompressible Navier–Stokes for CG and HDG discretizations are deduced. In the case of CG, total \mathbf{ndof} is sum of velocity DOFs, $\mathbf{ndof}(u)$, approximated with degree k and pressure DOFs, $\mathbf{ndof}(p)$, approximated with degree $k - 1$. In the case of HDG, pressure DOFs reduce to single unknown per each element unlike in CG. The expressions for total \mathbf{ndof} in the case of CG (with static condensation) and HDG can be written as follows,

$$\begin{aligned}\mathbf{ndof}_c &= d \mathbf{ndof}(u) + \mathbf{ndof}(p), \\ \mathbf{ndof}_h &= d \mathbf{ndof}(\hat{u}) + \mathbf{ndof}(\rho),\end{aligned}\tag{B.2}$$

where \mathbf{ndof}_c and \mathbf{ndof}_h are the total number of DOFs in CG (with static condensation) and HDG discretizations, respectively and d is the spatial dimension of the problem. These expressions for \mathbf{ndof} can be readily extended to conjugate heat transfer problem by considering the temperature DOFs. For instance, the total number of DOFs in the case of conjugate heat transfer problem are,

$$\begin{aligned}\mathbf{ndof}_{cc} &= d \mathbf{ndof}(u) + \mathbf{ndof}(p) + \mathbf{ndof}(\theta_f) + \mathbf{ndof}(\theta_s), \\ \mathbf{ndof}_{ch} &= d \mathbf{ndof}(\hat{u}) + \mathbf{ndof}(\rho) + \mathbf{ndof}(\hat{\theta}_f) + \mathbf{ndof}(\theta_s), \\ \mathbf{ndof}_{hh} &= d \mathbf{ndof}(\hat{u}) + \mathbf{ndof}(\rho) + \mathbf{ndof}(\hat{\theta}_f) + \mathbf{ndof}(\hat{\theta}_s),\end{aligned}\tag{B.3}$$

where \mathbf{ndof}_{cc} , \mathbf{ndof}_{ch} and \mathbf{ndof}_{hh} are the total DOF count for CG-CG, CG-HDG and HDG-HDG discretizations, respectively.

Now, the number of non-zero entries is considered for the conjugate heat transfer problem. Since, Taylor–Hood approximations use the same degree for velocity and temperature, the blocks involving velocity and temperature are of same size. The \mathbf{nnz} count of the blocks involving velocity and pressure needs to be developed as they are approximated with different degrees in Taylor–Hood family of elements. Hence, total \mathbf{nnz} for conjugate heat transfer problem can be expressed as follows,

$$\begin{aligned}\mathbf{nnz}_{cc} &= d^2 \mathbf{nnz}(u_1 u_1) + 2d (\mathbf{nnz}(u_1 p) + \mathbf{nnz}(u_1 \theta_f)) + \mathbf{nnz}(p p) + \mathbf{nnz}(\theta_f \theta_f) + \mathbf{nnz}(\theta_s \theta_s), \\ \mathbf{nnz}_{ch} &= d^2 \mathbf{nnz}(\hat{u}_1 \hat{u}_1) + 2d (\mathbf{nnz}(\hat{u}_1 \rho) + \mathbf{nnz}(\hat{u}_1 \hat{\theta}_f)) + \mathbf{nnz}(\hat{\theta}_f \hat{\theta}_f) + \mathbf{nnz}(\theta_s \theta_s), \\ \mathbf{nnz}_{hh} &= d^2 \mathbf{nnz}(\hat{u}_1 \hat{u}_1) + 2d (\mathbf{nnz}(\hat{u}_1 \rho) + \mathbf{nnz}(\hat{u}_1 \hat{\theta}_f)) + \mathbf{nnz}(\hat{\theta}_f \hat{\theta}_f) + \mathbf{nnz}(\hat{\theta}_s \hat{\theta}_s),\end{aligned}\tag{B.4}$$

where \mathbf{nnz}_{cc} , \mathbf{nnz}_{ch} and \mathbf{nnz}_{hh} corresponds to CG-CG, CG-HDG and HDG-HDG discretizations, respectively.

The expressions developed by Huerta *et al.* (2013) can be applied to each component in equations (B.2), (B.3) and (B.4). The expressions for \mathbf{ndof} and \mathbf{nnz} for a scalar unknown system (Laplace equation) with CG (with static condensation) and HDG discretizations are presented as follows,

$$\begin{aligned}\text{CG:} \quad \mathbf{ndof} &= \sum_{i=0}^{d-1} n_i \mathbf{ndof}_i^{int}, \quad \mathbf{nnz} = \sum_{i=0}^{d-1} c_i n_i \mathbf{ndof}_i^{int}, \\ \text{HDG:} \quad \mathbf{ndof} &= n_{d-1} \mathbf{ndof}_{d-1}, \quad \mathbf{nnz} = c n_{d-1} \mathbf{ndof}_{d-1},\end{aligned}\tag{B.5}$$

where the parameters are defined as follows,

d : Spatial dimension of the problem.

c_i : The number of nodes that are connected to a global node that is on the interior of an i -dimensional entity for a CG discretization. 0-dimensional entity corresponds to vertex nodes, 1-dimensional is edge, 2-dimensional and 3-dimensional entities are polygon and cell, respectively.

Table B.1: Total number of nodes and interior nodes in a i -dimensional entity for simplices and parallelotopes.

Dimension	Simplices		Parallelotopes	
	ndof_i	$\text{ndof}_i^{\text{int}}$	ndof_i	$\text{ndof}_i^{\text{int}}$
0	1	1	1	1
1	$k + 1$	$k - 1$	$k + 1$	$k - 1$
2	$(k + 1)(k + 2)/2$	$(k - 1)(k - 2)/2$	$(k + 1)^2$	$(k - 1)^2$
3	$(k + 1)(k + 2)(k + 3)/2$	$(k - 1)(k - 2)(k - 3)/2$	$(k + 1)^3$	$(k - 1)^3$

Table B.2: Expressions of the total number of connected nodes to a global node of i -dimensional entity in CG discretization.

Simplices	c_i	Expression	Parallelotopes	c_i	Expression
Triangles	c_0	$3k^2 + 3k + 1$	Quads	c_0	$4k^2 + 4k + 1$
	c_1	$k^2 + 2k + 1$		c_1	$2k^2 + 3k + 1$
	c_2	$(k^2 + 3k + 2)/2$		c_2	$k^2 + 2k + 1$
Tets	c_0	$4k^3 + 6k^2 + 4k + 1$	Hexes	c_0	$8k^3 + 12k^2 + 6k + 1$
	c_1	$(6k^3 + 18k^2 + 19k + 7)/7$		c_1	$4k^3 + 8k^2 + 5k + 1$
	c_2	$(k^3 + 3k^2 + 13k + 6)/6$		c_2	$2k^3 + 5k^2 + 4k + 1$
	c_3	$(k^3 + 6k^2 + 11k + 6)/6$		c_3	$k^3 + 3k^2 + 3k + 1$

n_i : The number of i -dimensional entities in the mesh. This is already presented in table 2.1 in terms of number of elements, n_{el} .

$\text{ndof}_i^{\text{int}}$: The number of interior nodes of the i -dimensional entity. It can be expressed in terms of degree of approximation, k . Table B.1 gives this information.

c : The number of nodes that are connected to a global node in HDG discretization.

ndof_i : The number of nodes of the i -dimensional entity. It can also be expressed in terms of degree of approximation, k . This is also presented in table B.1.

The formulae are self-explanatory, for instance, there are n_0 number of vertices in a mesh ($\text{ndof}_0^{\text{int}} = 1$) and each vertex has connectivities with c_0 number of other vertices. Hence, there will be $c_0 n_0 \text{ndof}_0^{\text{int}}$ number of entries in the global stiffness matrix. Similarly, there are $n_1 \text{ndof}_1^{\text{int}}$ number of interior nodes on all the edges of a mesh and each node is connected to c_1 other nodes. Finally, the summation of $c_0 n_0 + c_1 n_1 \text{ndof}_1^{\text{int}}$ gives **nnz** count in a statically condensed CG discretization. By taking the summation till d instead of $d - 1$, *i.e.*, taking the contributions from interior nodes of the elements by summing $c_2 n_2 \text{ndof}_2^{\text{int}}$ gives the **nnz** count in a CG discretization without static condensation. Using the similar arguments, the summation $n_0 + n_1 \text{ndof}_1^{\text{int}}$ corresponds to sum of interior nodes of vertices and edges in a mesh which gives the number of nodes of the mesh. In a scalar unknown system, total number of nodes in the mesh is the total number of DOFs. Table B.2 gives the expressions for ndof_i , $\text{ndof}_i^{\text{int}}$ and c_i , respectively in terms of degree of approximation, k , for CG discretization.

The connectivities is based on edges/faces in the HDG depending on spatial dimension. In 2-D, there are n_1 number of edges and $n_1 \text{ndof}_1$ number of nodes in the mesh. Each node is connected to c other nodes and hence, the number of entries are $c_1 n_1 \text{ndof}_1$. Table B.3

Table B.3: Expressions of the total number of connected nodes to a global node of i -dimensional entity in HDG discretization.

Simplices	c	Parallelotopes	c
Triangles	$5k + 5$	Quads	$7k + 7$
Tets	$(7k^2 + 21k + 14)/2$	Hexes	$11k^2 + 22k + 11$

Table B.4: Expressions to deduce the **ndof** for coupled models considered in the work.

Coupled model		Expression
CG(T_{k+1})-CG($T_{k+1}T_k$)	ndof (u)	$\sum_{i=0}^{d-1} n_i \text{ndof}_i^{\text{int}}(k+1)$
	ndof (p)	$\sum_{i=0}^{d-1} n_i \text{ndof}_i^{\text{int}}(k)$
	ndof (θ_f)	$\sum_{i=0}^{d-1} n_i \text{ndof}_i^{\text{int}}(k+1)$
	ndof (θ_s)	$\sum_{i=0}^{d-1} n_i^s \text{ndof}_i^{\text{int}}(k+1)$
CG(T_{k+1})-HDG(T_k)	ndof (\hat{u})	$n_{d-1} \text{ndof}_{d-1}(k)$
	ndof (ρ)	n_{el}
	ndof ($\hat{\theta}_f$)	$n_{d-1} \text{ndof}_{d-1}(k)$
	ndof (θ_s)	$\sum_{i=0}^{d-1} n_i^s \text{ndof}_i^{\text{int}}(k+1)$
HDG(T_k)-HDG(T_k)	ndof (\hat{u})	$n_{d-1} \text{ndof}_{d-1}(k)$
	ndof (ρ)	n_{el}
	ndof ($\hat{\theta}_f$)	$n_{d-1} \text{ndof}_{d-1}(k)$
	ndof ($\hat{\theta}_s$)	$n_{d-1} \text{ndof}_{d-1}(k)$

has the expressions for c in terms of k for HDG discretization. The number of mean pressure DOFs, **ndof**(ρ) is equal to number of elements, n_{el} , in the mesh.

If $\text{ndof}_i^{\text{int}}(k)$, $\text{ndof}_i(k)$ are the number of interior nodes and total number of nodes (interior+boundary) of the i -dimensional entity for degree of approximation k , the components in equation (B.3) are presented in table B.4. The number of entities of the solid domain are denoted by n_i^s in the expressions provided in the table B.4. Hence, the total **ndof** depends on the number of elements in fluid (Ω_D) and solid (Ω_C) domains. Deducing the **ndof** count for incompressible Navier–Stokes equations presented in table 2.2 is straightforward using equation (B.2), table B.4 and the auxiliary data provided.

Turning to number of non-zero entries now, first CG-CG coupled model is considered in deriving the total **nnz** count. As stated earlier, **nnz** count for each block involving velocity and pressure is approximately the same. If a coupled CG(T_{k+1})-CG($T_{k+1}T_k$) model is considered, **nnz**($u_1 u_1$)/**nnz**($u_1 \theta_f$)/**nnz**($\theta_f \theta_f$)/**nnz**($\theta_s \theta_s$) count can be obtained by using equation (B.5) and replacing k by $k+1$ in the expressions presented in tables B.2 and B.1. Similarly, **nnz**(pp) is computed by using the degree of approximation, k . The blocks containing velocity and pressure needs special attention as they are both approximated with different degree. Each global pressure node has c_i number of connected i -dimensional velocity node entities. Hence, the connectivities should be computed based on the number of connected velocity nodes to each global pressure node. The expressions for c_i in this case is the same as in the case of ($u_1 u_1$) block.

In the case of CG-HDG and HDG-HDG discretizations, the **nnz** count for velocity and

Table B.5: Expressions for $\mathbf{nnz}(\hat{u}_1 \rho)$ in terms of k and number of elements, n_{el} .

	$\mathbf{nnz}(\hat{u}_1 \rho)$	$\mathbf{nnz}(\hat{u}_1 \rho)$ in terms of k
Triangular	$3 \mathbf{ndof}_1 n_{el}$	$(3k + 3) n_{el}$
Quads	$4 \mathbf{ndof}_1 n_{el}$	$(4k + 4) n_{el}$
Tets	$4 \mathbf{ndof}_2 n_{el}$	$(2k^2 + 4k + 2) n_{el}$
Hexes	$6 \mathbf{ndof}_2 n_{el}$	$(6k^2 + 12k + 6) n_{el}$

Table B.6: Expressions for \mathbf{nnz} for several blocks of unknowns for considered coupled models in the present work.

Coupled model		Expression
CG(T_{k+1})-CG($T_{k+1}T_k$)	$\mathbf{nnz}(u_1 u_1)$	$\sum_{i=0}^{d-1} c_i(k+1) n_i \mathbf{ndof}_i^{int}(k+1)$
	$\mathbf{nnz}(u_1 p)$	$\sum_{i=0}^{d-1} c_i(k+1) n_i \mathbf{ndof}_i^{int}(k)$
	$\mathbf{nnz}(p p)$	$\sum_{i=0}^{d-1} c_i(k) n_i \mathbf{ndof}_i^{int}(k)$
	$\mathbf{nnz}(u_1 \theta_f)$	$\sum_{i=0}^{d-1} c_i(k+1) n_i \mathbf{ndof}_i^{int}(k+1)$
	$\mathbf{nnz}(\theta_f \theta_f)$	$\sum_{i=0}^{d-1} c_i(k+1) n_i \mathbf{ndof}_i^{int}(k+1)$
	$\mathbf{nnz}(\theta_s \theta_s)$	$\sum_{i=0}^{d-1} c_i(k+1) n_i^s \mathbf{ndof}_i^{int}(k+1)$
CG(T_{k+1})-HDG(T_k)	$\mathbf{nnz}(\hat{u}_1 \hat{u}_1)$	$c(k) n_{d-1} \mathbf{ndof}_{d-1}(k)$
	$\mathbf{nnz}(\hat{u}_1 \rho)$	$n_s \mathbf{ndof}_{d-1}(k) n_{el}$
	$\mathbf{nnz}(\hat{u}_1 \hat{\theta}_f)$	$c(k) n_{d-1} \mathbf{ndof}_{d-1}(k)$
	$\mathbf{nnz}(\hat{\theta}_f \hat{\theta}_f)$	$c(k) n_{d-1} \mathbf{ndof}_{d-1}(k)$
	$\mathbf{nnz}(\theta_s \theta_s)$	$\sum_{i=0}^{d-1} c_i(k+1) n_i^s \mathbf{ndof}_i^{int}(k+1)$
HDG(T_k)-HDG(T_k)	$\mathbf{nnz}(\hat{u}_1 \hat{u}_1)$	$c(k) n_{d-1} \mathbf{ndof}_{d-1}(k)$
	$\mathbf{nnz}(\hat{u}_1 \rho)$	$n_s \mathbf{ndof}_{d-1}(k) n_{el}$
	$\mathbf{nnz}(\hat{u}_1 \hat{\theta}_f)$	$c(k) n_{d-1} \mathbf{ndof}_{d-1}(k)$
	$\mathbf{nnz}(\hat{\theta}_f \hat{\theta}_f)$	$c(k) n_{d-1} \mathbf{ndof}_{d-1}(k)$
	$\mathbf{nnz}(\hat{\theta}_s \hat{\theta}_s)$	$c(k) n_{d-1} \mathbf{ndof}_{d-1}(k)$

temperature DOFs can be readily computed using the relations provided in table B.3. The major difference in counting \mathbf{nnz} comes from pressure DOFs compared to CG discretization. It is explained earlier that in HDG discretization of Stokes/Navier–Stokes equations, the pressure unknowns reduces to one scalar constant per element. Each pressure unknown, ρ_e , is connected to 3(4) edges in triangular(quadrilateral) elements in 2-D and 4(6) faces in tetrahedral(hexahedral) elements in 3-D. Table B.5 presents the $\mathbf{nnz}(\hat{u}_1 \rho)$ for different types of elements in 2-D and 3-D.

If $c_i(k)$ is the number of connected nodes of i -dimensional entity for degree of approximation k and $\mathbf{ndof}_i^{int}(k)$ is the number of interior nodes of i -dimensional entity for degree of approximation k , the \mathbf{nnz} count for blocks $(u_1 u_1)$ and $(u_1 p)$ can be expressed as shown in table B.6. As indicated earlier, n_i^s in expressions in table B.6 corresponds to the number of entities in the solid domain. Another new parameter introduced here is n_s , which represents number of edges (faces) in 2-D (3-D) elements. The parameter n_s assumes a value of $d + 1$ for simplices and $d + 2$ for parallelotopes as presented in table B.5.

The theoretical count on **nnz** for all the considered discretization can now be obtained by substituting the expressions for each block in (B.4). The final expressions are presented in table 3.5 and the expressions are verified using the numerical results of conjugate heat transfer problem. It is observed that the formulae are within 5% of the actual **nnz** count for all the discretizations.

Appendix C

Definition of elemental matrices in CG and HDG discretizations

In this appendix, the elemental matrices that arise from all the CG and HDG discretizations discussed before are defined. All the variables presented in this section are the elemental variables. Variable $\mathbf{L}^{(e)}$ is a second-order tensor and it is represented as a column vector, $[l_{11} \ l_{12} \ l_{21} \ l_{22}]^{(e)T}$, in the numerical computations.

The independent variables $(\mathbf{L}^{(e)}, \mathbf{u}^{(e)}, p^{(e)}, \theta^{(e)}, \mathbf{q}^{(e)}, \hat{\mathbf{u}}^{(e)}, \rho^{(e)}, \hat{\theta}^{(e)})$ over each element, Ω^e , can be approximated as follows,

$$\begin{aligned} \mathbf{L}^{(e)}(\boldsymbol{\xi}) &= \boldsymbol{\psi}_L(\boldsymbol{\xi})\mathbf{L}^{(e)}, \quad \mathbf{u}^{(e)}(\boldsymbol{\xi}) = \boldsymbol{\psi}_u(\boldsymbol{\xi})\mathbf{u}^{(e)}, \quad p^{(e)}(\boldsymbol{\xi}) = \boldsymbol{\psi}_p(\boldsymbol{\xi})p^{(e)} && \text{in } \Omega^e, \\ \theta^{(e)}(\boldsymbol{\xi}) &= \boldsymbol{\psi}_\theta(\boldsymbol{\xi})\theta^{(e)}, \quad \mathbf{q}^{(e)}(\boldsymbol{\xi}) = \boldsymbol{\psi}_q(\boldsymbol{\xi})\mathbf{q}^{(e)} && \text{in } \Omega^e, \\ \hat{\mathbf{u}}^{(e)}(\boldsymbol{\xi}) &= \boldsymbol{\psi}_{\hat{\mathbf{u}}}(\boldsymbol{\xi})\hat{\mathbf{u}}^{(e)}, \quad \hat{\theta}^{(e)}(\boldsymbol{\xi}) = \boldsymbol{\psi}_{\hat{\theta}}(\boldsymbol{\xi})\hat{\theta}^{(e)} && \text{on } \partial\Omega^e, \end{aligned} \quad (\text{C.1})$$

where $\boldsymbol{\psi}_L(\boldsymbol{\xi})$, $\boldsymbol{\psi}_u(\boldsymbol{\xi})$, $\boldsymbol{\psi}_p(\boldsymbol{\xi})$, $\boldsymbol{\psi}_\theta(\boldsymbol{\xi})$, $\boldsymbol{\psi}_q(\boldsymbol{\xi})$, $\boldsymbol{\psi}_{\hat{\mathbf{u}}}(\boldsymbol{\xi})$ and $\boldsymbol{\psi}_{\hat{\theta}}(\boldsymbol{\xi})$ are matrices that gather the approximation functions of respective unknowns, while $\mathbf{L}^{(e)}$, $\mathbf{u}^{(e)}$, $p^{(e)}$, $\theta^{(e)}$, $\mathbf{q}^{(e)}$, $\hat{\mathbf{u}}^{(e)}$ and $\hat{\theta}^{(e)}$ are the elemental nodal column vectors of gradient of velocity, velocity, pressure, temperature, flux, velocity trace and temperature trace, respectively. $\boldsymbol{\xi}$ and ξ represent the coordinate in the area and line reference domains, respectively. $\hat{\mathbf{u}}^{(e)}$ contains the trace of velocity on each face of the element and it can be represented as $[\hat{\mathbf{u}}^{\mathbf{F}_{e1}} \dots \hat{\mathbf{u}}^{\mathbf{F}_{en}}]^T$, where \mathbf{F}_{ef} is the f^{th} face of e^{th} element. Here, $n = 3$ in the case of triangular elements, while $n = 4$ for quadrilateral elements. From now on explicit dependence on $\boldsymbol{\xi}$ and ξ will be omitted for the sake of simplicity. The approximation functions can be represented as follows,

$$\begin{aligned} \boldsymbol{\psi}_L &= \begin{bmatrix} \boldsymbol{\psi} & & & \\ & \boldsymbol{\psi} & & \\ & & \boldsymbol{\psi} & \\ & & & \boldsymbol{\psi} \end{bmatrix}, \quad \boldsymbol{\psi}_u = \boldsymbol{\psi}_q = \begin{bmatrix} \boldsymbol{\psi} & \\ & \boldsymbol{\psi} \end{bmatrix}, \quad \boldsymbol{\psi}_p = \boldsymbol{\psi}_\theta = \boldsymbol{\psi}, \\ \boldsymbol{\psi}_{\hat{\mathbf{u}}} &= \begin{bmatrix} \boldsymbol{\psi}_{\mathbf{F}_{e1}} & \dots & \boldsymbol{\psi}_{\mathbf{F}_{en}} \\ & \boldsymbol{\psi}_{\mathbf{F}_{e1}} & \dots & \boldsymbol{\psi}_{\mathbf{F}_{en}} \end{bmatrix}, \quad \boldsymbol{\psi}_{\hat{\theta}} = \begin{bmatrix} \boldsymbol{\psi}_{\mathbf{F}_{e1}} & \dots & \boldsymbol{\psi}_{\mathbf{F}_{en}} \end{bmatrix}, \end{aligned} \quad (\text{C.2})$$

where $\boldsymbol{\psi}$ is the matrix that gathers the shape functions associated to the nodes of the elements and $\boldsymbol{\psi}_{\mathbf{F}_{ef}}$ is the matrix collecting the shape functions associated to the nodes along the sides of the element. The shape function matrices corresponding to CG discretization is represented

using an over-line notation, *i.e.*, matrix containing the shape functions of velocity is denoted by $\bar{\boldsymbol{\psi}}_u$.

Some notation used to represent the element matrices in case of both HDG and CG is given as follows,

$$\tilde{\nabla} \equiv \begin{bmatrix} \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} & & \\ & \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} & \\ & & & \end{bmatrix}, \quad \tilde{\mathbf{N}} \equiv \begin{bmatrix} n_1 & n_2 & & \\ & & n_1 & n_2 \end{bmatrix}. \quad (\text{C.3})$$

The elemental matrices corresponding to the discretization of Navier–Stokes equations for CG presented in (2.4) are as follows,

$$\begin{aligned} \mathbf{K}^{(e)} &= \left(\left(\tilde{\nabla}^T \bar{\boldsymbol{\psi}}_u \right)^T, \nu \tilde{\nabla}^T \bar{\boldsymbol{\psi}}_u \right)_{\Omega^e}, & \mathbf{C}^{(e)}(\mathbf{u}) &= \left(\bar{\boldsymbol{\psi}}_u^T \mathbf{u}, \tilde{\nabla}^T \bar{\boldsymbol{\psi}}_u \right)_{\Omega^e}, \\ \mathbf{G}^{(e)} &= - \left((\nabla \bar{\boldsymbol{\psi}}_p)^T, \bar{\boldsymbol{\psi}}_u \right)_{\Omega^e}, & \bar{\mathbf{s}}^{(e)} &= \left(\bar{\boldsymbol{\psi}}_u^T, \bar{\mathbf{s}} \right)_{\Omega^e}, \\ \bar{\mathbf{t}}^{(e)} &= \langle \bar{\boldsymbol{\psi}}_u^T, \bar{\mathbf{t}} \rangle_{\partial\Omega_N^e}. \end{aligned} \quad (\text{C.4})$$

Similarly, the elemental matrices from HDG discretization of Navier–Stokes equations (2.14) are given as follows,

$$\begin{aligned} \mathbf{A}_{LL}^{(e)} &= (\boldsymbol{\psi}_L^T, \boldsymbol{\psi}_L)_{\Omega^e}, & \mathbf{A}_{Lu}^{(e)} &= \left((\tilde{\nabla} \boldsymbol{\psi}_L)^T, \boldsymbol{\psi}_u \right)_{\Omega^e}, \\ \mathbf{A}_{L\hat{u}}^{(e)} &= - \langle (\tilde{\mathbf{N}} \boldsymbol{\psi}_L)^T, \boldsymbol{\psi}_{\hat{u}} \rangle_{\partial\Omega^e}. \end{aligned} \quad (\text{C.5})$$

The non-linear convective matrices in equation (2.9b) can be expressed as follows,

$$\begin{aligned} \mathbf{C}_{uu}^{(e)}(\mathbf{u}) &= - \left[\begin{array}{cc} \left(\boldsymbol{\psi}_{u,1}^T, u_1 \boldsymbol{\psi}_u \right)_{\Omega^e} & \left(\boldsymbol{\psi}_{u,2}^T, u_1 \boldsymbol{\psi}_u \right)_{\Omega^e} \\ \left(\boldsymbol{\psi}_{u,1}^T, u_2 \boldsymbol{\psi}_u \right)_{\Omega^e} & \left(\boldsymbol{\psi}_{u,2}^T, u_2 \boldsymbol{\psi}_u \right)_{\Omega^e} \end{array} \right], \\ \mathbf{C}_{u\hat{u}}^{(e)}(\hat{\mathbf{u}}) &= \left[\begin{array}{cc} \langle \boldsymbol{\psi}_u^T, \hat{u}_1 \boldsymbol{\psi}_{\hat{u}} n_1 \rangle_{\partial\Omega^e} & \langle \boldsymbol{\psi}_u^T, \hat{u}_1 \boldsymbol{\psi}_{\hat{u}} n_2 \rangle_{\partial\Omega^e} \\ \langle \boldsymbol{\psi}_u^T, \hat{u}_2 \boldsymbol{\psi}_{\hat{u}} n_1 \rangle_{\partial\Omega^e} & \langle \boldsymbol{\psi}_u^T, \hat{u}_2 \boldsymbol{\psi}_{\hat{u}} n_2 \rangle_{\partial\Omega^e} \end{array} \right]. \end{aligned} \quad (\text{C.6})$$

The discretization of rest of the terms in equation (2.9b) results in the following,

$$\begin{aligned} \mathbf{A}_{uL}^{(e)} &= - \left(\boldsymbol{\psi}_u^T, \nu \tilde{\nabla} \boldsymbol{\psi}_L \right)_{\Omega^e}, & \mathbf{A}_{up}^{(e)} &= (\boldsymbol{\psi}_u^T, \nabla \boldsymbol{\psi}_p)_{\Omega^e}, \\ \mathbf{A}_{uu}^{(e)} &= \langle \boldsymbol{\psi}_u^T, \tau_u \boldsymbol{\psi}_u \rangle_{\partial\Omega^e}, & \mathbf{A}_{u\hat{u}}^{(e)} &= - \langle \boldsymbol{\psi}_u^T, \tau_u \boldsymbol{\psi}_{\hat{u}} \rangle_{\partial\Omega^e}, \\ \bar{\mathbf{s}}^{(e)} &= (\boldsymbol{\psi}_u^T, \bar{\mathbf{s}})_{\Omega^e}. \end{aligned} \quad (\text{C.7})$$

The matrices of the continuity equation (2.9c) and constraint (3.5b) are presented as follows,

$$\begin{aligned} \mathbf{A}_{pu}^{(e)} &= - \left((\nabla \boldsymbol{\psi}_p)^T, \boldsymbol{\psi}_u \right)_{\Omega^e}, & \mathbf{A}_{p\hat{u}}^{(e)} &= \langle (\mathbf{n} \boldsymbol{\psi}_p)^T, \boldsymbol{\psi}_{\hat{u}} \rangle_{\partial\Omega^e}, \\ \mathbf{A}_{\rho p}^{(e)} &= \frac{1}{|\partial\Omega^e|} \langle \boldsymbol{\psi}_p, 1 \rangle_{\partial\Omega^e}. \end{aligned} \quad (\text{C.8})$$

Finally, the elemental matrices of global problem (2.12a) and (2.12b) can be expressed as,

$$\begin{aligned} \mathbf{A}_{\hat{u}p}^{(e)} &= - \langle \boldsymbol{\psi}_{\hat{u}}^T, (\mathbf{n} \boldsymbol{\psi}_p) \rangle_{\partial\Omega^e}, & \mathbf{A}_{\hat{u}L}^{(e)} &= \langle \boldsymbol{\psi}_{\hat{u}}^T, (\tilde{\mathbf{N}} \boldsymbol{\psi}_L) \rangle_{\partial\Omega^e}, \\ \mathbf{A}_{\hat{u}\hat{u}}^{(e)} &= \langle \boldsymbol{\psi}_{\hat{u}}^T, \tau_u \boldsymbol{\psi}_{\hat{u}} \rangle_{\partial\Omega^e}, & \mathbf{A}_{\hat{u}u}^{(e)} &= - \langle \boldsymbol{\psi}_{\hat{u}}^T, \tau_u \boldsymbol{\psi}_u \rangle_{\partial\Omega^e}, \\ \mathbf{A}_{\rho\hat{u}}^{(e)} &= \langle \mathbf{n}^T \boldsymbol{\psi}_{\hat{u}}, 1 \rangle_{\partial\Omega^e}, & \bar{\mathbf{t}}^{(e)} &= \langle \boldsymbol{\psi}_{\hat{u}}^T, \bar{\mathbf{t}} \rangle_{\partial\Omega_N^e}. \end{aligned} \quad (\text{C.9})$$

The tangent operator terms, necessary within the Newton–Raphson iterative method, associated with the residual (2.14) are trivial, except for the ones associated with $\mathbf{C}_{uu}^{(e)}(\mathbf{u})$ and $\mathbf{C}_{u\hat{u}}^{(e)}(\hat{\mathbf{u}})$ matrices, given by (C.6). These are expressed by

$$\mathbf{C}_{Tuu}^{(e)} = \begin{bmatrix} \left(\psi_{u,1}^T, u_1 \psi_u \right)_{\Omega^e} + \left(\psi_{u,2}^T, u_2 \psi_u \right)_{\Omega^e} & \mathbf{0} \\ \mathbf{0} & \left(\psi_{u,1}^T, u_1 \psi_u \right)_{\Omega^e} + \left(\psi_{u,2}^T, u_2 \psi_u \right)_{\Omega^e} \end{bmatrix}, \quad (\text{C.10})$$

$$\mathbf{C}_{Tu\hat{u}}^{(e)} = \begin{bmatrix} \langle \psi_u^T, (\hat{\mathbf{u}} \cdot \mathbf{n}) \psi_{\hat{u}} \rangle_{\partial\Omega^e} & \mathbf{0} \\ \mathbf{0} & \langle \psi_u^T, (\hat{\mathbf{u}} \cdot \mathbf{n}) \psi_{\hat{u}} \rangle_{\partial\Omega^e} \end{bmatrix}. \quad (\text{C.11})$$

The definition of elemental matrices that arise from the coupled Navier–Stokes/convection–diffusion equations using HDG discretization presented in (3.10) are,

$$\begin{aligned} \mathbf{A}_{qq}^{(e)} &= (\psi_q^T, \alpha^{-1} \psi_q)_{\Omega^e}, & \mathbf{A}_{q\theta}^{(e)} &= - \left((\tilde{\nabla}^T \psi_q)^T, \psi_\theta \right)_{\Omega^e}, \\ \mathbf{A}_{q\hat{\theta}}^{(e)} &= \langle (\mathbf{n}^T \psi_q)^T, \psi_{\hat{\theta}} \rangle_{\partial\Omega^e}, & \mathbf{A}_{\theta q}^{(e)} &= \left(\psi_\theta^T, (\tilde{\nabla}^T \psi_q)^T \right)_{\Omega^e}, \\ \mathbf{A}_{\theta\hat{\theta}}^{(e)} &= \langle \psi_\theta^T, \tau_\theta \psi_{\hat{\theta}} \rangle_{\partial\Omega^e}, & \mathbf{A}_{\hat{\theta}\hat{\theta}}^{(e)} &= - \langle \psi_\theta^T, \tau_\theta \psi_{\hat{\theta}} \rangle_{\partial\Omega^e}, \\ \mathbf{A}_{\hat{\theta}\theta}^{(e)} &= - \langle \psi_\theta^T, \tau_\theta \psi_{\hat{\theta}} \rangle_{\partial\Omega^e}, & \mathbf{A}_{\theta\theta}^{(e)} &= \langle \psi_\theta^T, \tau_\theta \psi_\theta \rangle_{\partial\Omega^e}, \\ \mathbf{A}_{\hat{\theta}q}^{(e)} &= \langle \psi_\theta^T, (\mathbf{n}^T \psi_q) \rangle_{\partial\Omega^e}, & \bar{\mathbf{g}}^{(e)} &= (\psi_\theta^T, \bar{g})_{\Omega^e}, \\ \mathbf{A}_{u\theta}^{(e)} &= (\psi_u^T, \mathbf{g} \beta \psi_\theta)_{\Omega^e}, & \mathbf{f}_{\theta 0}^{(e)} &= (\psi_u^T, \mathbf{g} \beta \theta_0)_{\Omega^e}. \end{aligned} \quad (\text{C.12})$$

The non-linear matrices from the system in (3.10) are defined as follows,

$$\begin{aligned} \mathbf{C}_{\theta\theta}^{(e)}(\mathbf{u}) &= - \langle \psi_{\theta,1}^T, u_1 \psi_\theta \rangle_{\partial\Omega^e} - \langle \psi_{\theta,2}^T, u_2 \psi_\theta \rangle_{\partial\Omega^e}, \\ \mathbf{C}_{\theta\hat{\theta}}^{(e)}(\hat{\mathbf{u}}) &= \langle \psi_\theta^T, (\hat{\mathbf{u}} \cdot \mathbf{n}) \psi_{\hat{\theta}} \rangle_{\partial\Omega^e}, \\ \mathbf{C}_{\hat{\theta}\hat{\theta}}^{(e)}(\hat{\mathbf{u}}) &= \langle \psi_\theta^T, (\hat{\mathbf{u}} \cdot \mathbf{n}) \psi_{\hat{\theta}} \rangle_{\partial\Omega^e}. \end{aligned} \quad (\text{C.13})$$

The tangent operators of the non-linear matrices already presented in (C.13) are given as,

$$\begin{aligned} \mathbf{C}_{\theta\theta T}^{(e)}(\theta) &= - \left[\langle \psi_{\theta,1}^T, \theta \psi_\theta \rangle_{\partial\Omega^e} \quad \langle \psi_{\theta,2}^T, \theta \psi_\theta \rangle_{\partial\Omega^e} \right], \\ \mathbf{C}_{\theta\hat{\theta}T}^{(e)}(\hat{\theta}) &= \left[\langle \psi_\theta^T, \hat{\theta} n_1 \psi_{\hat{u}} \rangle_{\partial\Omega^e} \quad \langle \psi_\theta^T, \hat{\theta} n_2 \psi_{\hat{u}} \rangle_{\partial\Omega^e} \right], \\ \mathbf{C}_{\hat{\theta}\hat{\theta}T}^{(e)}(\hat{\theta}) &= \left[\langle \psi_\theta^T, \hat{\theta} n_1 \psi_{\hat{u}} \rangle_{\partial\Omega^e} \quad \langle \psi_\theta^T, \hat{\theta} n_2 \psi_{\hat{u}} \rangle_{\partial\Omega^e} \right]. \end{aligned} \quad (\text{C.14})$$

The matrices that arise from the coupling of HDG and CG on the interface, Γ_I , in the equation (3.30) can be expressed as follows,

$$\begin{aligned} \mathbf{B}_{\theta\theta}^{(e)} &= - \langle \psi_\theta^T, \tau \bar{\psi}_\theta \rangle_{\partial\Omega^e \cap \Gamma_I}, & \mathbf{B}_{q\theta}^{(e)} &= \langle (\mathbf{n}^T \psi_q)^T, \bar{\psi}_\theta \rangle_{\partial\Omega^e \cap \Gamma_I} \\ \mathbf{B}_{\theta q}^{(e)} &= - \mathbf{B}_{q\theta}^{(e)T}, & \mathbf{K}_{\theta\theta}^{(e)} &= ((\nabla \bar{\psi}_\theta)^T, \alpha_2 \nabla \bar{\psi}_\theta)_{\Omega_2^e}. \end{aligned} \quad (\text{C.15})$$

The mass matrices from the time discretization in (4.12) can be expressed as follows,

$$\begin{aligned} \mathbf{M}_{uu}^{(e)} &= (\psi_u^T, \psi_u)_{\Omega_f^e}, & \mathbf{M}_{\theta\theta}^{(e)} &= (\psi_\theta^T, \psi_\theta)_{\Omega_f^e}, \\ \bar{\mathbf{M}}_{\theta\theta}^{(e)} &= (\bar{\psi}_\theta^T, \bar{\psi}_\theta)_{\Omega_s^e}. \end{aligned} \quad (\text{C.16})$$

Finally, the elemental matrices and body force vectors corresponding to the discretization in solid part, Ω_s , and interface, Γ_R , in (4.12) are presented in (C.17).

$$\begin{aligned}
\mathbf{K}_{RR}^{(k)} &= \begin{cases} \left\langle \frac{1}{1-\epsilon} \boldsymbol{\psi}_R^{(e)T}, \boldsymbol{\psi}_R^{(e)} \right\rangle_{\Gamma_R^e} & \text{if } k = e, \\ - \left\langle \boldsymbol{\psi}_R^{(e)T}, \left\langle \boldsymbol{\psi}_R^{(k)}, \frac{\cos \gamma^{(e)} \cos \gamma^{(k)}}{2r} \right\rangle_{\Gamma_R^{(k)}} \right\rangle_{\Gamma_R^e} & \text{if } k \neq e, \end{cases} \\
\mathbf{f}_\theta^{(e)} &= \left\langle \bar{\boldsymbol{\psi}}_\theta^T, h_a(\theta_s) \bar{\boldsymbol{\psi}}_\theta(\boldsymbol{\theta}_a - \boldsymbol{\theta}_s) \right\rangle_{\Gamma_t^e \cup \Gamma_b^e} + \left\langle \epsilon \sigma \bar{\boldsymbol{\psi}}_\theta^T, ((\bar{\boldsymbol{\psi}}_\theta \boldsymbol{\theta}_a)^4 - (\bar{\boldsymbol{\psi}}_\theta \boldsymbol{\theta}_s)^4) \right\rangle_{\Gamma_t^e \cup \Gamma_b^e} \\
&\quad - \left\langle \frac{\epsilon}{1-\epsilon} \bar{\boldsymbol{\psi}}_\theta^T, (\sigma (\bar{\boldsymbol{\psi}}_\theta \boldsymbol{\theta}_s)^4 - \boldsymbol{\psi}_R \mathbf{R}) \right\rangle_{\Gamma_R^e}, \\
\mathbf{f}_R^{(k)} &= \begin{cases} \left\langle \sigma \frac{\epsilon}{1-\epsilon} \boldsymbol{\psi}_R^T, (\bar{\boldsymbol{\psi}}_\theta \boldsymbol{\theta}_s)^4 \right\rangle_{\Gamma_R^e} & \text{if } k = e \\ \mathbf{0} & \text{if } k \neq e \end{cases}
\end{aligned} \tag{C.17}$$

In (C.17), $\boldsymbol{\psi}_R$ denote the matrix consisting 1-D nodal shape functions of the radiosity variable. It is assumed in this work that the degree of approximation of radiosity is the same as that of temperature in the solid domain. The linearisation and tangent operators of the radiosity equation are discussed in-detail in López (2017).

Bibliography

- Abramowitz, M. (1974). *Handbook of Mathematical Functions, With Formulas, Graphs, and Mathematical Tables*. Dover Publications, Incorporated.
- Akin, J. E. (2005). *Finite Element Analysis with Error Estimators: An Introduction to the FEM and Adaptive Error Analysis for Engineering Students*. Elsevier Science.
- Alotto, P., A. Bertoni, I. Perugia and D. Schötzau (2001). Discontinuous finite element methods for the simulation of rotating electrical machines. *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering*, 20(2), 448–462.
- Amestoy, P. R., T. A. Davis and I. S. Duff (1996). An approximate minimum degree ordering algorithm. *SIAM Journal on Matrix Analysis and Applications*, 17(4), 886–905.
- Arioli, M., J. W. Demmel and I. S. Duff (1989). Solving sparse linear systems with sparse backward error. *SIAM Journal on Matrix Analysis and Applications*, 10(2), 165–190.
- Arnold, D. N., Brezzi F. (1985). Mixed and nonconforming finite element methods: implementation, postprocessing and error estimates. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 19(1), 7–32.
- Arnold, D. N. (1982). An interior penalty finite element method with discontinuous elements. *SIAM Journal on Numerical Analysis*, 19(4), 742–760.
- Arnold, D. N., F. Brezzi, B. Cockburn and D. Marini (2000). *Discontinuous Galerkin Methods for Elliptic Problems*, pages 89–101. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Atkins, H. L. and C. Shu (1996). Quadrature-free implementation of the discontinuous Galerkin method for hyperbolic equations. Technical report.
- Babuška, I. and W. C. Rheinboldt (1978). A-posteriori error estimates for the finite element method. *International Journal for Numerical Methods in Engineering*, 12(10), 1597–1615.
- Bassi, F. and S. Rebay (1997). A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. *Journal of Computational Physics*, 131(2), 267 – 279.
- Baumann, C. E. and J. T. Oden (1999). A discontinuous *hp* finite element method for convection-diffusion problems. *Computer Methods in Applied Mechanics and Engineering*, 175(3), 311 – 341.

- Berrone, S. (2001). Adaptive discretization of stationary and incompressible Navier–Stokes equations by stabilized finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 190(34), 4435 – 4455.
- Betts, P. L. and V. Haroutunian (1983). A stream function finite element solution for two-dimensional natural convection with accurate representation of Nusselt number variations near a corner. *International Journal for Numerical Methods in Fluids*, 3(6), 605–622.
- Blain, C. A. and T. C. Massey (2005). Application of a coupled discontinuous-continuous Galerkin finite element shallow water model to coastal ocean dynamics. *Ocean Modelling*, 10(3-4), 283 – 315.
- Braza, M., P. Chassaing and H. Ha Minh (1986). Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder. *Journal of Fluid Mechanics*, 165, 79–130.
- Brenan, K., S. Campbell and L. Petzold (1995). *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics.
- Brezzi, F., J. Douglas and L. D. Marini (1985). Two families of mixed finite elements for second order elliptic problems. *Numerische Mathematik*, 47(2), 217–235.
- Brezzi, F. and M. Fortin (1991). *Mixed and Hybrid Finite Element Methods*. Springer Series in Computational Mathematics. Springer-Verlag, New York.
- Calvo, M., T. Grande and R. D. Grigorieff (1990). On the zero stability of the variable order variable stepsize BDF-Formulas. *Numerische Mathematik*, 57(1), 39–50.
- Cangiani, A., J. Chapman, E. Georgoulis and M. Jensen (2013). On the Stability of Continuous-Discontinuous Galerkin Methods for Advection-Diffusion-Reaction Problems. *Journal of Scientific Computing*, 57(2), 313–330.
- Cangiani, A., E. H. Georgoulis and M. Jensen (2006). Continuous and discontinuous finite element methods for convection-diffusion problems: A comparison. Technical report, Dept. of Mathematics, University of Leicester.
- Celiker, F., B. Cockburn and K. Shi (2010). Hybridizable Discontinuous Galerkin methods for Timoshenko beams. *Journal of Scientific Computing*, 44(1), 1–37.
- Cesmelioglu, A., B. Cockburn, N. C. Nguyen and J. Peraire (2013). Analysis of HDG methods for Oseen equations. *Journal of Scientific Computing*, 55(2), 392–431.
- Cesmelioglu, A., B. Cockburn and W. Qiu (2017). Analysis of a hybridizable discontinuous Galerkin method for the steady-state incompressible Navier–Stokes equations. *Mathematics of Computation*, 86(306), 1643–1670.
- Chavent, G. and G. Salzano (1982). A finite-element method for the 1-D water flooding problem with gravity. *Journal of Computational Physics*, 45(3), 307 – 344.
- Chen, Y. and B. Cockburn (2012). Analysis of variable-degree HDG methods for convection–diffusion equations. Part I: general nonconforming meshes. *IMA Journal of Numerical Analysis*, 32(4), 1267–1293.

- Chen, Y. and B. Cockburn (2014). Analysis of variable-degree HDG methods for convection–diffusion equations. Part II: Semimatching nonconforming meshes. *Mathematics of Computation*, 83(285), 87–111.
- Cockburn, B., B. Dong, J. Guzman, M. Restelli and R. Sacco (2009a). A hybridizable discontinuous Galerkin method for steady-state convection-diffusion-reaction problems. *SIAM Journal on Scientific Computing*, 31(5), 3827–3846.
- Cockburn, B. and J. Gopalakrishnan (2005). New hybridization techniques. *GAMM-Mitteilungen*, 28(2), 154–182.
- Cockburn, B. and J. Gopalakrishnan (2009). The derivation of hybridizable discontinuous Galerkin methods for Stokes flow. *SIAM Journal on Numerical Analysis*, 47(2), 1092–1125.
- Cockburn, B., J. Gopalakrishnan and R. Lazarov (2009b). Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. *SIAM Journal on Numerical Analysis*, 47(2), 1319–1365.
- Cockburn, B., J. Gopalakrishnan, N. C. Nguyen, J. Peraire and F. J. Sayas (2011). Analysis of HDG methods for Stokes flow. *Math. Comput.*, 80(274), 723–760.
- Cockburn, B., F. Li and C. Shu (2004). Locally divergence-free discontinuous Galerkin methods for the Maxwell equations. *Journal of Computational Physics*, 194(2), 588 – 610.
- Cockburn, B. and K. Mustapha (2015). A hybridizable discontinuous Galerkin method for fractional diffusion problems. *Numerische Mathematik*, 130(2), 293–314.
- Cockburn, B., N. C. Nguyen and J. Peraire (2010). A comparison of HDG methods for Stokes flow. *Journal of Scientific Computing*, 45(1), 215–237.
- Cockburn, B. and K. Shi (2014). Devising methods for Stokes flow: An overview. *Computers & Fluids*, 98, 221 – 229. 12th USNCCM mini-symposium of High-Order Methods for Computational Fluid Dynamics - A special issue dedicated to the 80th birthday of Professor Antony Jameson.
- Cockburn, B. and C. Shu (1998). The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis*, 35(6), 2440–2463.
- Correia, J. R., F. A. Branco, J. G. Ferreira, Y. Bai and T. Keller (2010). Fire protection systems for building floors made of pultruded GFRP profiles: Part 1: Experimental investigations. *Composites Part B: Engineering*, 41(8), 617 – 629.
- Courant, R. (1943). Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of the American Mathematical Society*, 49(1), 1–23. Also available in the Classic Reprints Series, International Journal for Numerical Methods in Engineering, volume 37, issue 13, pages 2159–2187, 1994.
- Cousins, B. R., S. L. Borne, A. Linke, L. G. Rebholz and Z. Wang (2013). Efficient linear solvers for incompressible flow simulations using Scott–Vogelius finite elements. *Numerical Methods for Partial Differential Equations*, 29(4), 1217–1237.

- Dawson, C. and J. Proft (2002a). Coupling of continuous and discontinuous Galerkin methods for transport problems. *Computer Methods in Applied Mechanics and Engineering*, 191(29-30), 3213 – 3231.
- Dawson, C. and J. Proft (2002b). Discontinuous and coupled continuous/discontinuous Galerkin methods for the shallow water equations. *Computer Methods in Applied Mechanics and Engineering*, 191(41-42), 4721 – 4746.
- Dawson, C. and J. Proft (2003). Discontinuous/continuous Galerkin methods for coupling the primitive and wave continuity equations of shallow water. *Computer Methods in Applied Mechanics and Engineering*, 192(47 - 48), 5123 – 5145.
- Dawson, C. and J. Proft (2004). Coupled discontinuous and continuous Galerkin finite element methods for the depth-integrated shallow water equations. *Computer Methods in Applied Mechanics and Engineering*, 193(3 - 5), 289 – 318.
- De Vahl Davis, G. (1983). Natural convection of air in a square cavity: A bench mark numerical solution. *International Journal for Numerical Methods in Fluids*, 3(3), 249–264.
- Devloo, P. R. B., T. Forti and S. M. Gomes (2007). A combined continuous-discontinuous finite element method for convection-diffusion problems. *Latin American Journal of Solids and Structures*, 4(3), 229–246.
- Dorfman, A. S. (2009). *Conjugate Problems in Convective Heat Transfer*. CRC Press Taylor & Francis.
- Eça, L. and M. Hoekstra (2014). A procedure for the estimation of the numerical uncertainty of CFD calculations based on grid refinement studies. *Journal of Computational Physics*, 262, 104 – 130.
- Engsig-Karup, A. P., J. S. Hesthaven, H. B. Bingham and P. A. Madsen (2006). Nodal DG-FEM solution of high-order Boussinesq-type equations. *Journal of Engineering Mathematics*, 56(3), 351–370.
- Estep, D. J., M. G. Larson and R. D. Williams (2000). *Estimating the Error of Numerical Solutions of Systems of Reaction-diffusion Equations*. Number no. 696 in American Mathematical Society: Memoirs of the American Mathematical Society. American Mathematical Society.
- Fernandez, P., N.C. Nguyen and J. Peraire (2017). The hybridized discontinuous Galerkin method for Implicit Large-Eddy simulation of transitional turbulent flows. *Journal of Computational Physics*, 336, 308 – 329.
- Florio, J., J. B. Henderson, F. L. Test and R. Hariharan (1991). A study of the effects of the assumption of local-thermal equilibrium on the overall thermally-induced response of a decomposing, glass-filled polymer composite. *International Journal of Heat and Mass Transfer*, 34(1), 135 – 147.
- Forti, D. and L. Dedè (2015). Semi-implicit BDF time discretization of the Navier–Stokes equations with VMS-LES modeling in a High Performance Computing framework. *Computers & Fluids*, 117, 168 – 182.

- Fraeijis de Veubeke, B. (1965). *Displacement and equilibrium models in finite element method, of stress analysis*. Wiley, New York.
- Fu, G., B. Cockburn and H. Stolarski (2015a). Analysis of an HDG method for linear elasticity. *International Journal for Numerical Methods in Engineering*, 102(3-4), 551–575.
- Fu, G., W. Qiu and W. Zhang (2015b). An analysis of HDG methods for convection-dominated diffusion problems. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 49(1), 225–256.
- Gatica, G. N. and F. A. Sequeira (2016). A Priori and a Posteriori error analyses of an augmented HDG method for a class of Quasi-Newtonian Stokes flows. *Journal of Scientific Computing*, 69(3), 1192–1250.
- Gear, C. (1971). Simultaneous Numerical Solution of Differential-Algebraic Equations. *IEEE Transactions on Circuit Theory*, 18(1), 89–95.
- Georgoulis, E. H. and E. Süli (2005). Optimal error estimates for the hp-version interior penalty discontinuous Galerkin finite element method. *IMA Journal of Numerical Analysis*, 25(1), 205.
- Geuzaine, C. and J.-F. Remacle (2009). Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11), 1309–1331.
- Giorgiani, G., S. Fernández-Méndez and A. Huerta (2014). Hybridizable Discontinuous Galerkin with degree adaptivity for the incompressible Navier–Stokes equations. *Computers & Fluids*, 98, 196 – 208. 12th USNCCM mini-symposium of High-Order Methods for Computational Fluid Dynamics - A special issue dedicated to the 80th birthday of Professor Antony Jameson.
- Giorgiani, G., D. Modesto, S. Fernández-Méndez and A. Huerta (2013). High-order continuous and discontinuous Galerkin methods for wave problems. *International Journal for Numerical Methods in Fluids*, 73(10), 883–903.
- Grigorieff, R. D. (1983). Stability of multistep-methods on variable grids. *Numerische Mathematik*, 42(3), 359–377.
- Guglielmi, N. and M. Zennaro (2001). On the zero-stability of variable stepsize multistep methods: the spectral radius approach. *Numerische Mathematik*, 88(3), 445–458.
- Guo, H., Q. Zhang and Y. Yang (2014). A combined mixed finite element method and local discontinuous Galerkin method for miscible displacement problem in porous media. *Science China Mathematics*, 57(11), 2301–2320.
- Gürkan, C., M. Kronbichler and S. Fernández-Méndez (2017). eXtended Hybridizable Discontinuous Galerkin with Heaviside Enrichment for Heat Bimaterial problems. *Journal of Scientific Computing*, 72(2), 542–567.
- Gürkan, C., E. Sala-Lardies, M. Kronbichler and S. Fernández-Méndez (2016). eXtended hybridizable discontinuous Galerkin (X-HDG) for void problems. *Journal of Scientific Computing*, 66(3), 1313–1333.

- Güzey, S., B. Cockburn and H. K. Stolarski (2007). The embedded discontinuous Galerkin method: application to linear shell problems. *International Journal for Numerical Methods in Engineering*, 70(7), 757–790.
- Hairer, E., S. P. Nørsett and G. Wanner (1993). *Solving Ordinary Differential Equations I (2Nd Revised. Ed.): Nonstiff Problems*. Springer-Verlag New York, Inc., New York, NY, USA.
- Henderson, J. B., J. A. Wiebelt and M. R. Tant (1985). A Model for the Thermal Response of Polymer Composite Materials with Experimental Verification. *Journal of Composite Materials*, 19(6), 579–595.
- Hesthaven, J. S. and T. Warburton (2002). Nodal high-order methods on unstructured grids: I. time-domain solution of Maxwell’s equations. *Journal of Computational Physics*, 181(1), 186 – 221.
- Hesthaven, J. S. and T. Warburton (2007). *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Springer Publishing Company, Incorporated, 1st edition.
- Holzapfel, G. A. (2000). *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*. John Wiley & Sons.
- HSL (2016). A collection of Fortran codes for large scale scientific computation.
- Huerta, A., A. Angeloski, X. Roca and J. Peraire (2013). Efficiency of high-order elements for continuous and discontinuous Galerkin methods. *International Journal for Numerical Methods in Engineering*, 96(9), 529–560.
- Hughes, T. J. R., G. Scovazzi, P. B. Bochev and A. Buffa (2006). A multiscale discontinuous Galerkin method with the computational structure of a continuous Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 195(19-22), 2761 – 2787.
- ISO 834 (1975). *Fire resistance tests. Elements of building construction*. International Standards Organization, Genève.
- Kaminski, D. A. and C. Prakash (1986). Conjugate natural convection in a square enclosure: effect of conduction in one of the vertical walls. *International Journal of Heat and Mass Transfer*, 29(12), 1979 – 1988.
- Keith, J. G., L. Alexander, G. R. Leo and E. W. Nicholas (2012). Stabilizing poor mass conservation in incompressible flow problems with large irrotational forcing and application to thermal convection. *Computer Methods in Applied Mechanics and Engineering*, 237 – 240(Supplement C), 166 – 176.
- Keller, T., C. Tracy and A. Zhou (2006). Structural response of liquid-cooled GFRP slabs subjected to fire - Part II: Thermo-chemical and thermo-mechanical modeling. *Composites Part A: Applied Science and Manufacturing*, 37(9), 1296 – 1308.
- Kirby, R. M., S. J. Sherwin and B. Cockburn (2012). To CG or to HDG: A comparative study. *Journal of Scientific Computing*, 51(1), 183–212.

- Kovasznyai, L. I. G. (1948). Laminar flow behind a two-dimensional grid. *Mathematical Proceedings of the Cambridge Philosophical Society*, 44, 58–62.
- LaCaN (2016). EZ4U. Mesh generation environment.
- Lambert, J. D. (1991). *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*. Wiley, first edition.
- Le Quéré, P. (1991). Accurate solutions to the square thermally driven cavity at high Rayleigh number. *Computers & Fluids*, 20(1), 29 – 41.
- Lesaint, P. and P. A. Raviart (1974). On a finite element method for solving the neutron transport equation. *Publications mathématiques et informatique de Rennes*, (S4), 1–40.
- Li, L., S. Lanteri and R. Perrussel (2014). A Hybridizable Discontinuous Galerkin Method Combined to a Schwarz Algorithm for the Solution of 3D Time-harmonic Maxwell’s Equation. *Journal of Computational Physics*, 256, 563–581.
- Liu, C., X. Zheng and C. H. Sung (1998). Preconditioned Multigrid Methods for Unsteady Incompressible Flows. *Journal of Computational Physics*, 139(1), 35 – 57.
- Liu, R., M. F. Wheeler, C. N. Dawson and R. H. Dean (2009). On a coupled discontinuous/-continuous Galerkin framework and an adaptive penalty scheme for poroelasticity problems. *Computer Methods in Applied Mechanics and Engineering*, 198(41-44), 3499 – 3510.
- Looyeh, M. R. E. and P. Bettess (1998). A finite element model for the fire-performance of GFRP panels including variable thermal properties. *Finite Elements in Analysis and Design*, 30(4), 313 – 324.
- López, C. (2017). *Numerical modelling of the thermomechanical behaviour of GFRP pultruded profiles subjected to fire*. Ph.D. thesis, Instituto Superior Técnico, Universidade de Lisboa.
- López, C., C. Tiago, J. R. Correia and J. A. Teixeira de Freitas (2014). Application of the Finite Element Method to the coupled heat transfer/enclosure radiation/fluid dynamics problem. Technical Report DTC 03/2014, ICIST.
- Lubich, Ch. and M. Roche (1990). Rosenbrock methods for differential-algebraic systems with solution-dependent singular matrix multiplying the derivative. *Computing*, 43(4), 325–342.
- Maday, Y., A. T. Patera and E. M. Rønquist (1990). An operator-integration-factor splitting method for time-dependent problems: Application to incompressible fluid flow. *Journal of Scientific Computing*, 5(4), 263–292.
- Miano, V. U. and A. G. Gibson (2009). Fire model for fibre reinforced plastic composites using apparent thermal diffusivity (ATD). *Plastics, Rubber and Composites*, 38(2-4), 87–92.
- Miller, J. R. and P. M. Weaver (2003). Temperature profiles in composite plates subject to time-dependent complex boundary conditions. *Composite Structures*, 59(2), 267 – 278.
- Misra, D. and A. Sarkar (1997). Finite element analysis of conjugate natural convection in a square enclosure with a conducting vertical wall. *Computer Methods in Applied Mechanics and Engineering*, 141(3), 205 – 219.

- Montlaur, A., S. Fernández-Méndez and A. Huerta (2012). High-order implicit time integration for unsteady incompressible flows. *International Journal for Numerical Methods in Fluids*, 70(5), 603–626.
- Morgado, T. M., J. R. Correia and F. A. Branco (2013a). Fire protection systems for glass fibre reinforced polymer (GFRP) pultruded profiles. Fire resistance tests on GFRP pultruded columns. Internal report IST-5-5.
- Morgado, T. M., F. Nunes, J. R. Correia and F. A. Branco (2013b). Fire protection systems for glass fibre reinforced polymer (GFRP) pultruded profiles. Fire resistance tests on GFRP pultruded beams. Internal report IST-5-4.
- Moro, D., N. C. Nguyen and J. Peraire (2011). Navier–Stokes solution using Hybridizable Discontinuous Galerkin methods. In *20th AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics.
- Mouritz, A. P., S. Feih, E. Kandare, Z. Mathys, A. G. Gibson, P. E. Des Jardin, S. W. Case and B. Y. Lattimer (2009). Review of fire structural modelling of polymer composites. *Composites Part A: Applied Science and Manufacturing*, 40(12), 1800 – 1814. Special Issue: CompTest 2008.
- Nguyen, N. C. and J. Peraire (2012). Hybridizable discontinuous Galerkin methods for partial differential equations in continuum mechanics. *Journal of Computational Physics*, 231(18), 5955 – 5988.
- Nguyen, N. C., J. Peraire and B. Cockburn (2009a). An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection–diffusion equations. *Journal of Computational Physics*, 228(23), 8841 – 8855.
- Nguyen, N. C., J. Peraire and B. Cockburn (2010). A hybridizable discontinuous Galerkin method for Stokes flow. *Computer Methods in Applied Mechanics and Engineering*, 199(9–12), 582 – 597.
- Nguyen, N. C., J. Peraire and B. Cockburn (2011a). High-order implicit hybridizable discontinuous Galerkin methods for acoustics and elastodynamics. *Journal of Computational Physics*, 230(10), 3695 – 3718.
- Nguyen, N. C., J. Peraire and B. Cockburn (2011b). Hybridizable discontinuous Galerkin methods for the time-harmonic Maxwell equations. *Journal of Computational Physics*, 230(19), 7151 – 7175.
- Nguyen, N. C., J. Peraire and B. Cockburn (2011c). An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 230(4), 1147 – 1170.
- Nguyen, N.C., J. Peraire and B. Cockburn (2009b). An implicit high-order hybridizable discontinuous Galerkin method for linear convection–diffusion equations. *Journal of Computational Physics*, 228(9), 3232 – 3254.
- Nitsche, J. (1971). Über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 36(1), 9–15.

- Oberkampf, W. L. and C. J. Roy (2010). *Verification and Validation in Scientific Computing*. Cambridge University Press, New York, NY, USA, 1st edition.
- Paipuri, M., S. Fernández-Méndez and C. Tiago (2017a). Comparison of continuous and hybridizable discontinuous Galerkin methods in incompressible fluid flow problems. In Irene Arias, Jesús María Blanco, Stephane Clain, Paulo Flores, Paulo Lourenço, Juan José Ródenas and Manuel Tur (Editors) *Congreso de Métodos Numéricos en Ingeniería – CMN 2017*. International Center for Numerical Methods in Engineering (CIMNE), Valencia, Spain.
- Paipuri, M., C. Tiago and S. Fernández-Méndez (2017b). Coupling of continuous and hybridizable discontinuous Galerkin methods: Application to conjugate heat transfer problem. *Journal of Scientific computing*. (Submitted).
- Peraire, J., N. C. Nguyen and B. Cockburn (2010). A hybridizable discontinuous Galerkin method for the compressible Euler and Navier–Stokes equations.
- Peraire, J. and P.-O. Persson (2008). The compact discontinuous Galerkin (CDG) method for elliptic problems. *SIAM Journal on Scientific Computing*, 30(4), 1806–1824.
- Perugia, I. and D. Schötzau (2001). On the coupling of local discontinuous Galerkin and conforming finite element methods. *Journal of Scientific Computing*, 16(4), 411–433.
- Petra, C. G., O. Schenk and M. Anitescu (2014a). Real-time stochastic optimization of complex energy systems on high-performance computers. *IEEE Computing in Science & Engineering*, 16(5), 32–42.
- Petra, C. G., O. Schenk, M. Lubin and K. Gärtner (2014b). An augmented incomplete factorization approach for computing the Schur complement in stochastic optimization. *SIAM Journal on Scientific Computing*, 36(2), C139–C162.
- Prada, D. (2016). *A hybridizable discontinuous Galerkin method for nonlinear porous media viscoelasticity with applications in ophthalmology*. Ph.D. thesis, Purdue University, Purdue e-Pubs.
- Qiu, W. and K. Shi (2016). An HDG method for convection diffusion equation. *Journal of Scientific Computing*, 66(1), 346–357.
- Rajani, B.N., A. Kandasamy and S. Majumdar (2009). Numerical simulation of laminar flow past a circular cylinder. *Applied Mathematical Modelling*, 33(3), 1228 – 1247.
- Rang, J. (2007). Design of DIRK schemes for solving the Navier–Stokes equations. *Informatik-Berichte der Technischen Universität Braunschweig*, 2007-02.
- Rang, J. (2016). High order time discretisation methods for incompressible Navier–Stokes equations. *PAMM: Proceedings in Applied Mathematics and Mechanics*, 16(1), 759–760.
- Rang, J. and L. Angermann (2005). New Rosenbrock W-Methods of Order 3 for Partial Differential Algebraic Equations of Index 1. *BIT Numerical Mathematics*, 45(4), 761–787.
- Raviart, P. A. and J. M. Thomas (1977). *A mixed finite element method for 2-nd order elliptic problems*, pages 292–315. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Reed, W. H. and T. R. Hill (1973). Triangular mesh methods for the neutron transport equation. *Los Alamos Report LA-UR-73-479*.

- Rhebergen, S., B. Cockburn and J. J. W. van der Vegt (2013). A space-time discontinuous Galerkin method for the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 233, 339 – 358.
- Richards, S. A. (1997). Completed Richardson extrapolation in space and time. *Communications in Numerical Methods in Engineering*, 13(7), 573–582.
- Richardson, L. F. (1911). The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations, with an Application to the Stresses in a Masonry Dam. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 210(459-470), 307–357.
- Rivière, B., M. F. Wheeler and V. Girault (2001). A priori error estimates for finite element methods based on discontinuous approximation spaces for elliptic problems. *SIAM Journal on Numerical Analysis*, 39(3), 902–931.
- Roache, P. J. (2009). *Fundamentals of verification and validation*. Hermosa Publisher.
- Roca, X., C. Nguyen and J. Peraire (2013). Scalable parallelization of the hybridized discontinuous Galerkin method for compressible flow. In *21st AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics.
- Sarrate, J. and A. Huerta (2000). Efficient unstructured quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, 49(10), 1327–1350.
- Sarrate, J. and A. Huerta (2002). Generación automática de mallas no estructuradas y formadas exclusivamente por cuadriláteros sobre superficies curvas en \mathbb{R}^3 . *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, 18(1), 79–93.
- Scott, L. R. and M. Vogelius (1985). Norm estimates for a maximal right inverse of the divergence operator in spaces of piecewise polynomials. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 19(1), 111–143.
- Sevilla, R. and A. Huerta (2016). *Tutorial on Hybridizable Discontinuous Galerkin (HDG) for Second-Order Elliptic Problems*, pages 105–129. Springer International Publishing.
- Sheldon, J. P., S. T. Miller and J. S. Pitt (2016). A hybridizable discontinuous Galerkin method for modeling fluid-structure interaction. *Journal of Computational Physics*, 326, 91 – 114.
- Soon, S.-C., B. Cockburn and H. K. Stolarski (2009). A hybridizable discontinuous Galerkin method for linear elasticity. *International Journal for Numerical Methods in Engineering*, 80(8), 1058–1092.
- Steinebach, G. (1995). *Order reduction of ROW methods for DAEs and method of lines applications*. Preprint. Techn. Hochsch., Fachbereich Mathematik.
- Stella, F. and E. Bucchignani (1999). Rayleigh–Bénard Convection In Limited Domains: Part 1 - Oscillatory Flow. *Numerical Heat Transfer, Part A: Applications*, 36(1), 1–16.
- Sullivan, R. M. and N. J. Salamon (1992a). A finite element method for the thermochemical decomposition of polymeric materials-I. theory. *International Journal of Engineering Science*, 30(4), 431 – 441.

- Sullivan, R. M. and N. J. Salamon (1992b). A finite element method for the thermochemical decomposition of polymeric materials-II. carbon phenolic composites. *International Journal of Engineering Science*, 30(7), 939 – 951.
- Taylor, C. and P. Hood (1973). A numerical solution of the Navier–Stokes equations using the finite element technique. *Computers & Fluids*, 1(1), 73–100.
- Taylor, M. A., B. A. Wingate and R. E. Vincent (2000). An algorithm for computing Fekete points in the triangle. *SIAM Journal on Numerical Analysis*, 38(5), 1707–1720.
- Tracy, C. (2005). *Fire endurance of multicellular panels in an FRP building system*. Ph.D. thesis, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.
- Vaz, G., A. Koop, F. Pereira, H. Abreu and L. Eça (2016). Validation: What, why and how. In *ASME 2016 35th International Conference on Ocean, Offshore and Arctic Engineering*, volume 2. ASME, Busan, South Korea.
- Wang, B. and B.C. Khoo (2013). Hybridizable discontinuous Galerkin method (HDG) for Stokes interface flow. *Journal of Computational Physics*, 247, 262 – 278.
- Wang, Dong and Steven J. Ruuth (2008). Variable step-size implicit-explicit linear multi-step methods for time-dependent partial differential equations. *Journal of Computational Mathematics*, 26(6), 838–855.
- Wang, S., J. Yuan, W. Deng and Y. Wu (2016). A hybridized discontinuous Galerkin method for 2D fractional convection–diffusion equations. *Journal of Scientific Computing*, 68(2), 826–847.
- Whiteman, J. R. (1994). *The Mathematics of finite elements and applications: highlights 1993*. Chichester: Wiley.
- William, W. H. (1984). Condition estimates. *SIAM Journal on Scientific and Statistical Computing*, 5(2), 311–316.
- Yakovlev, S., D. Moxey, R. M. Kirby and S. J. Sherwin (2016). To CG or to HDG: A comparative study in 3D. *Journal of Scientific Computing*, 67(1), 192–220.
- Zarin, H. (2009). Continuous-discontinuous finite element method for convection-diffusion problems with characteristic layers. *Journal of Computational and Applied Mathematics*, 231(2), 626 – 636.
- Zhu, P., Z. Xie and S. Zhou (2011a). A uniformly convergent continuous-discontinuous Galerkin method for singularly perturbed problems of convection-diffusion type. *Applied Mathematics and Computation*, 217(9), 4781 – 4790.
- Zhu, P., X. Ziqing and Z. Shuzi (2011b). A coupled continuous-discontinuous FEM approach for convection diffusion equations. *Acta Mathematica Scientia*, 31(2), 601 – 612.